



TUGAS AKHIR - TE 145561

**PENGATURAN KECEPATAN MOTOR DC *SPINDLE* DENGAN
KONTROLER PI DAN PERANCANGAN MEKANIK MESIN CNC
*PORTABLE***

Rico Artha Dinata
NRP 2214030075

Dosen Pembimbing
1. Ir. Djoko Suprajitno Raharjo, M.T
2. Andri Ashfahani, ST., M.Sc

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***SPEED SETTING OF DC MOTOR SPINDLE WITH PI
CONTROLLER AND MECHANIC DESIGNING CNC PORTABLE
MACHINE***

Rico Artha Dinata
NRP 2214030075

Advisor

1. Ir. Djoko Suprajitno Raharjo, M.T
2. Andri Ashfahani, ST., M.Sc

***COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pengaturan Kecepatan Motor DC *Spindle* dengan Kontroler PI dan Perancangan Mekanik Mesin CNC *Portable***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 JULI 2017



Rico Artha Dinata
NRP 2214030075

-----Halaman ini sengaja dikosongkan-----

**PENGATURAN KECEPATAN MOTOR DC SPINDLE DENGAN
KONTROLER PI DAN PERANCANGAN MEKANIK MESIN
CNC PORTABLE**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Ir. Djoko Suprajitno Raharjo, MT
NIP. 195506221987011001

Andri Ashfahani, ST., M.Sc.
NIP. 2200201405003

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

PENGATURAN KECEPATAN MOTOR DC *SPINDLE* DENGAN KONTROLER PI DAN PERANCANGAN MEKANIK MESIN CNC *PORTABLE*

Nama : Rico Artha Dinata
NRP : 2214 030 075

Pembimbing 1: Ir. Djoko Suprajitno Raharjo, M.T
NIP : 195506221987011001

Pembimbing 2: Andri Ashfahani, ST., M.Sc
NIP : 2200201405003

ABSTRAK

Mesin CNC *Portable* 3 Axis merupakan mesin yang memiliki produktifitas tinggi, ketelitian pengerjaan tinggi dan dapat digabung dengan mesin lain. Dalam hal ini mesin dapat bekerja dengan perangkat tambahan sehingga pemakaian mesin CNC akan lebih efektif, waktu produksi lebih cepat, kapasitas produksi lebih besar, dan biaya pembuatan produk lebih murah. Keunggulan CNC *Portable* 3 Axis ini antara lain dapat di bawa kemana-mana, memiliki kemampuan sesuai mesin CNC pada umumnya serta dapat digunakan pada industri skala kecil seperti wirausaha pengerajin kayu. Penggunaan bahan besi dan akrilik memberikan kekuatan pada mesin sehingga mampu melakukan pengeboran dan pengukiran pada kayu. Pada tingkat kepresisian mesin dipengaruhi juga oleh motor *Spindle* yang digunakan pada bagian pengeboran. Motor *Spindle* yang dipakai berupa motor *Spindle* jenis DC *High Speed* dimana memiliki kecepatan putar tinggi. Pengaturan kecepatan motor DC *Spindle* menggunakan kontroler PI sehingga kecepatannya bisa stabil dan hasilnya lebih presisi.

Hasil dari tugas akhir ini mampu menggerakkan motor DC *Spindle* menggunakan *software* LabVIEW dan menyetabilkan kecepatan motor DC *Spindle* menggunakan kontroler PI. Selain itu juga membandingkan dengan penggunaan kontroler PID. Kemudian didapatkan parameter kontrol yang sesuai dan bisa cepat stabil yaitu $K_p = 0,012$ dan $\tau_i = 0.011$.

Kata Kunci: Mesin CNC *Portable*, Motor DC *Spindle*, Penggunaan bahan, Kontroler PI

-----Halaman ini sengaja dikosongkan-----

***SPEED SETTING OF DC MOTOR SPINDLE WITH PI
CONTROLLER AND MECHANIC DESIGNING CNC PORTABLE
MACHINE***

***Name : Rico Artha Dinata
Register Number : 2214 030 075***

***Advisor 1 : Ir. Djoko Suprajitno Raharjo, M.T
ID Number : 195506221987011001***

***Advisor 2 : Andri Ashfahani, ST., M.Sc
ID Number : 2200201405003***

ABSTRACT

Portable 3 Axis CNC machine is a machine that has high productivity, high accuracy and workmanship can be combined with other machines. In this case the machine can work with enhancements so that the use of CNC machines will be more effective, faster production time, greater production capacity, and cheaper product-making costs. The advantages of CNC Portable 3 Axis are, among others, can be brought anywhere, has the ability to fit CNC machines in general and can be used in small scale industries such as entrepreneurs wooden craftsmen. The use of iron and acrylic materials provide power to the engine so as to make drilling and engraving on wood. At the precision level of the engine is also affected by the Spindle motor used in the drilling section. Spindle motor is used in the form is type high speed DC Spindle motor which has high rotation speed. Setting the speed of a DC Spindle motor using a PI controller so the speed can be stable and the result is more precise.

The result of this final project is able to drive the DC Spindle motor using LabVIEW software and stabilize the speed of DC Spindle motor using PI controller. It also compares with the use of PID controllers. Then obtained the appropriate control parameters and can be quickly stable that is $K_p = 0.012$ and $\tau_f = 0.011$.

Keywords: CNC Portable Machine, DC Spindle Motor, Material Usage, PI Controller

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan laporan Tugas Akhir dengan judul:

PENGATURAN KECEPATAN MOTOR DC *SPINDLE* DENGAN KONTROLER PI DAN PERANCANGAN MEKANIK MESIN *CNC PORTABLE*

Penulis ingin menyampaikan terima kasih kepada beberapa pihak yang telah membantu menyusun dan menyelesaikan Tugas Akhir ini, diantaranya :

1. Kedua orang tua yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Bapak Ir. Djoko Suprajitno Raharjo, M.T selaku Dosen Pembimbing I dan Bapak Andri Ashfahani, ST., M.Sc selaku Dosen Pembimbing II yang telah membimbing penulis dalam menyelesaikan Tugas Akhir.
3. Para Dosen Penguji pada sidang Tugas Akhir penulis yang telah memberikan masukan yang penting guna menyempurnakan alat dan buku Tugas Akhir penulis.
4. Semua teman yang ada di Teknik D3 Teknik Elektro angkatan 2014 khususnya anggota kelompok Tugas Akhir Fahmi, Yoga Innes atas bantuan dan kerjasama yang telah diberikan.

Harapan besar penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan D3 Teknik Elektro pada khususnya. Penulis juga mengharapkan kritik dan saran atas buku Tugas Akhir ini karena penulis menyadari bahwa dalam penyusunan Tugas Akhir ini banyak terdapat kekurangan.

Surabaya, 19 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan	3
1.7 Relevansi	4
BAB II TEORI DASAR	5
2.1 Mesin CNC ^[6]	5
2.2 Motor DC ^[4]	6
2.2.1 Pengertian Motor DC	6
2.2.2 Pengaturan Kecepatan Motor DC	7
2.3 Komponen Penyusun Mesin CNC <i>Portable</i>	9
2.3.1 Komponen Mekanik	9
2.3.2 Komponen Elektrik	13
2.3.3 Komponen Kontroler	19
2.4 CAD/CAM	19
2.5 LabVIEW	20
2.5.1 <i>Front Panel</i>	20
2.5.2 <i>Diagram Block</i>	21
BAB III PERANCANGAN SISTEM KONTROL	23
3.1 Perancangan Pengaturan Kecepatan Motor DC <i>Spindle</i>	23

3.2 Perancangan Perangkat Lunak LabVIEW	28
3.3 Perancangan Kontroler PI pada Motor DC <i>Spindle</i>	34
3.3.1 Identifikasi <i>Plant</i> Motor DC <i>Spindle</i>	34
3.3.2 Efek Pengontrolan dengan Kontroler PI	36
3.3.3 <i>Trial and Error</i> Kontroler PI.....	39
3.4 Mekanik Mesin CNC	40
3.4.1 Komponen Pembentuk Mesin CNC	40
3.4.2 Dimensi dari Mesin CNC <i>Portable</i>	45
3.4.3 Tata Letak Komponen pada Mesin CNC <i>Portable</i> ...	47
BAB IV PENGUJIAN DAN ANALISA DATA	51
4.1 Pengujian Pergerakan Motor <i>Stepper</i>	51
4.2 Pengujian Kecepatan Motor pada LabVIEW	53
4.2.1 Pengujian Pengaturan Kecepatan Motor Tanpa Kontroler	55
4.2.2 Pengujian Pengaturan Kecepatan Motor dengan Kontroler PI.....	58
4.2.3 Pengujian Pengaturan Kecepatan Motor dengan Kontroler PID.....	68
4.2.4 Perbandingan Kontroler PI dan PID.....	73
BAB V KESIMPULAN DAN SARAN	77
5.1 Kesimpulan	77
5.2 Saran	77
DAFTAR PUSTAKA	79
LAMPIRAN A	81
LAMPIRAN B	83
B.1. <i>Datasheet Driver</i> Motor <i>Stepper</i> A4988	83
B.2. <i>Datasheet</i> Motor DC <i>Spindle</i>	87
LAMPIRAN C	89
DAFTAR RIWAYAT HIDUP	94

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Mekanisme Kerja Motor DC	6
Gambar 2.2 Gelombang PWM dengan <i>Duty Cycle</i> 50%	8
Gambar 2.3 Gelombang PWM.....	8
Gambar 2.4 Motor <i>Stepper Unipolar</i>	10
Gambar 2.5 Motor <i>Stepper Bipolar</i>	11
Gambar 2.6 Motor <i>Stepper</i>	11
Gambar 2.7 Motor <i>Spindle</i>	12
Gambar 2.8 <i>Ballscrew</i>	13
Gambar 2.9 Arduino UNO	16
Gambar 2.10 <i>Driver</i> Motor	16
Gambar 2.11 Skematik <i>Driver</i> Motor A988	17
Gambar 2.12 <i>Driver</i> Motor <i>Stepper</i> A4988	18
Gambar 2.13 Sensor Kecepatan	18
Gambar 2.14 <i>Power Supply</i>	19
Gambar 2.15 Tampilan <i>Front Panel</i> LabVIEW	21
Gambar 2.16 Tampilan <i>Control Palette</i> LabVIEW	21
Gambar 2.17 Tampilan <i>Diagram Block</i> LabVIEW	22
Gambar 2.18 Tampilan <i>Function Palette</i> LabVIEW	22
Gambar 3.1 Blok Fungsional Sistem Secara Keseluruhan	23
Gambar 3.2 Diagram Alur Perancangan Kendali Motor DC	23
Gambar 3.3 Arduino Uno.....	24
Gambar 3.4 <i>Pinout Driver</i> Motor DC <i>Spindle</i>	25
Gambar 3.5 Perancangan Pengaturan Kecepatan Motor DC <i>Spindle</i>	26
Gambar 3.6 <i>Flowchart</i> Sistem	27
Gambar 3.7 VISA <i>Resources Names</i>	29
Gambar 3.8 VISA <i>Configure Serial Port</i>	29
Gambar 3.9 Tampilan Komunikasi VISA pada <i>Diagram Block</i>	29
Gambar 3.10 Fungsi pada <i>While Loop</i>	30
Gambar 3.11 <i>Diagram Block</i> pada Masukan LabVIEW	31
Gambar 3.12 <i>Structures Case</i> Program LabVIEW	32
Gambar 3.13 Keseluruhan Perancangan Program LabVIEW	32
Gambar 3.14 Tampilan <i>Front Panel</i> Kecepatan Motor DC	33
Gambar 3.15 Tampilan <i>Diagram Block</i> Kecepatan Motor DC	33
Gambar 3.16 Program Sensor Kecepatan pada Arduino.....	35
Gambar 3.17 Grafik Respon Motor DC <i>Spindle</i>	36

Gambar 3.18 <i>Diagram Block</i> Kontroler P	37
Gambar 3.19 <i>Diagram Block</i> Kontroler I	38
Gambar 3.20 <i>Diagram Block</i> Kontroler PI	38
Gambar 3.21 Respon Sistem Terhadap Aksi Kontroler PI	39
Gambar 3.22 Program Kontrol PI di Arduino.....	39
Gambar 3.23 <i>Wiring Diagram Driver</i> Motor A4988 dengan Motor <i>Stepper</i>	41
Gambar 3.24 <i>Wiring Diagram Driver</i> Motor A4988	42
Gambar 3.25 Meja Kerja Mesin CNC <i>Portable</i>	44
Gambar 3.26 <i>Holder</i> Sumbu Z	44
Gambar 3.27 Desain Meja Kerja pada Mesin CNC <i>Portable</i>	45
Gambar 3.28 Desain <i>Column</i> Aksis Y	45
Gambar 3.29 Desain <i>Spindle Carriage</i> Aksis X	46
Gambar 3.30 Desain <i>Tool Holder</i> Aksis Z	46
Gambar 3.31 Desain Mesin CNC <i>Portable</i>	46
Gambar 3.32 Motor <i>Stepper</i> Sumbu Y Tampak Samping Kanan	47
Gambar 3.33 Motor <i>Stepper</i> Sumbu Y Tampak Samping Kiri	48
Gambar 3.34 Motor <i>Stepper</i> Sumbu X Tampak dari Belakang	48
Gambar 3.35 Motor <i>Stepper</i> Sumbu Z Tampak dari Tengah.....	49
Gambar 3.36 Motor DC <i>Spindle</i> pada Sumbu Z	49
Gambar 4.1 Grafik <i>Full Step</i>	52
Gambar 4.2 Grafik <i>Half Step</i>	53
Gambar 4.3 Tegangan dan Kecepatan Tanpa Kontroler PI	54
Gambar 4.4 Tegangan dan Kecepatan dengan Kontroler PI.....	55
Gambar 4.5 Komunikasi Serial VISA LabVIEW	55
Gambar 4.6 <i>Set Point</i> pada LabVIEW	56
Gambar 4.7 <i>Range</i> PWM.....	56
Gambar 4.8 <i>Set Point</i> 10.000 RPM.....	56
Gambar 4.9 <i>Set Point</i> 11.000 RPM.....	57
Gambar 4.10 <i>Set Point</i> 12.000 RPM.....	57
Gambar 4.11 Parameter Kp = 0,018 dan τi = 0.015 pada <i>Set Point</i> 10.000 RPM	59
Gambar 4.12 Parameter Kp = 0,018 dan τi = 0.015 pada <i>Set Point</i> 11.000 RPM	59
Gambar 4.13 Parameter Kp = 0,018 dan τi = 0.015 pada <i>Set Point</i> 12.000 RPM	60
Gambar 4.14 Parameter Kp = 0,012 dan τi = 0.011 pada <i>Set Point</i> 10.000 RPM	60

Gambar 4.15 Parameter $Kp = 0,012$ dan $\tau i = 0.011$ pada <i>Set Point</i> 11.000 RPM.....	61
Gambar 4.16 Parameter $Kp = 0,012$ dan $\tau i = 0.011$ pada <i>Set Point</i> 12.000 RPM.....	61
Gambar 4.17 Parameter $Kp = 0,026$ dan $\tau i = 0.020$ pada <i>Set Point</i> 10.000 RPM.....	62
Gambar 4.18 Parameter $Kp = 0,026$ dan $\tau i = 0.020$ pada <i>Set Point</i> 11.000 RPM.....	62
Gambar 4.19 Parameter $Kp = 0,026$ dan $\tau i = 0.020$ pada <i>Set Point</i> 12.000 RPM.....	63
Gambar 4.20 Parameter $Kp = 0,037$ dan $\tau i = 0.032$ pada <i>Set Point</i> 10.000 RPM.....	63
Gambar 4.21 Parameter $Kp = 0,037$ dan $\tau i = 0.032$ pada <i>Set Point</i> 11.000 RPM.....	64
Gambar 4.22 Parameter $Kp = 0,037$ dan $\tau i = 0.032$ pada <i>Set Point</i> 12.000 RPM.....	64
Gambar 4.23 Parameter $Kp = 0,047$ dan $\tau i = 0.046$ pada <i>Set Point</i> 10.000 RPM.....	65
Gambar 4.24 Parameter $Kp = 0,047$ dan $\tau i = 0.046$ pada <i>Set Point</i> 11.000 RPM.....	65
Gambar 4.25 Parameter $Kp = 0,047$ dan $\tau i = 0.046$ pada <i>Set Point</i> 12.000 RPM.....	66
Gambar 4.26 Parameter $Kp = 0,058$ dan $\tau i = 0.055$ pada <i>Set Point</i> 10.000 RPM.....	66
Gambar 4.27 Parameter $Kp = 0,058$ dan $\tau i = 0.055$ pada <i>Set Point</i> 11.000 RPM.....	67
Gambar 4.28 Parameter $Kp = 0,058$ dan $\tau i = 0.055$ pada <i>Set Point</i> 12.000 RPM.....	67
Gambar 4.29 Parameter $Kp = 0,018$, $\tau i = 0.015$, dan $\tau d = 0,001$ pada <i>Set Point</i> 12.000 RPM.....	69
Gambar 4.30 Parameter $Kp = 0,018$, $\tau i = 0.015$, dan $\tau d = 0,003$ pada <i>Set Point</i> 12.000 RPM.....	69
Gambar 4.31 Parameter $Kp = 0,018$, $\tau i = 0.015$, dan $\tau d = 0,005$ pada <i>Set Point</i> 12.000 RPM.....	70
Gambar 4.32 Parameter $Kp = 0,012$, $\tau i = 0.011$, dan $\tau d = 0,001$ pada <i>Set Point</i> 12.000 RPM.....	70
Gambar 4.33 Parameter $Kp = 0,012$, $\tau i = 0.011$, dan $\tau d = 0,003$ pada <i>Set Point</i> 12.000 RPM.....	71

Gambar 4.34 Parameter $Kp = 0,012$, $\tau i = 0.011$, dan $\tau d = 0,005$
pada *Set Point* 12.000 RPM71

Gambar 4.35 Parameter $Kp = 0,026$, $\tau i = 0.020$, dan $\tau d = 0,001$
pada *Set Point* 12.000 RPM72

Gambar 4.36 Parameter $Kp = 0,026$, $\tau i = 0.020$, dan $\tau d = 0,003$
pada *Set Point* 12.000 RPM72

Gambar 4.37 Parameter $Kp = 0,026$, $\tau i = 0.020$, dan $\tau d = 0,005$
pada *Set Point* 12.000 RPM73

Gambar 4.38 Kontroler PI pada *Set Point* 12.000 RPM74

Gambar 4.39 Kontroler PID pada *Set Point* 12.000 RPM74

DAFTAR TABEL

HALAMAN

Tabel 2.1 Data Teknis <i>Board</i> Arduino Uno	13
Tabel 2.2 <i>Micro Stepping Driver</i> Motor A4988.....	17
Tabel 3.1 Keterangan Konfigurasi Pin	26
Tabel 3.2 Efek dari Pengontrol P dan I	34
Tabel 3.3 Pin Konfigurasi <i>Driver</i> Motor A4988.....	41
Tabel 3.4 Data Teknis Motor DC <i>Spindle</i>	42
Tabel 3.5 Data Teknis <i>Driver</i> Motor DC <i>Spindle</i>	43
Tabel 4.1 Data <i>Full Step</i>	51
Tabel 4.2 Data <i>Half Step</i>	51
Tabel 4.3 Tegangan Motor Sebelum Pengontrolan PI	54
Tabel 4.4 Tegangan Motor Sesudah Pengontrolan PI	54
Tabel 4.5 Parameter Kontroler PI	58
Tabel 4.6 Parameter Kontroler PID.....	68

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman saat ini berpengaruh terhadap penemuan teknologi yang semakin maju. Terlebih di bidang industri, penggunaan komputer dan peralatan mekanik terbukti lebih efisien daripada bekerja secara manual menggunakan tenaga manusia. Perpaduan teknologi komputer dan peralatan mekanik ini menghasilkan suatu alat yang dinamakan CNC (*Computer Numerically Controlled*). Mesin CNC ini mampu bekerja secara otomatis sesuai yang kita inginkan sehingga hasil yang didapatkan lebih presisi daripada bekerja secara manual. Pemanfaatan teknologi ini berdampak terhadap pekerjaan manusia. Hal tersebut juga berimbas pula pada industri-industri kecil, menengah dan keatas, khususnya yang masih menggunakan peralatan konvensional atau bahkan masih menggunakan peralatan tradisional dan manual. Pemahaman teknologi secara mendasar dan mendalam dilakukan melalui pelaksanaan program yang tepat untuk memproduksi barang dan jasa sesuai dengan industrinya.

Mesin CNC yang terdapat di industri biasanya berukuran besar karena objek kerjanya berukuran besar pula. Penggunaan mesin CNC tidak semuanya efisien diterapkan di industri. Untuk skala industri menengah dan kecil, mesin tersebut tidaklah sesuai digunakan dikarenakan biayanya mahal dan peletakkannya yang terlalu besar sehingga memakan tempat yang besar. Persoalan ini dapat diatasi dengan membuat mesin CNC yang berukuran kecil sehingga dapat dibawa kemana-mana dan dapat dioperasikan kapanpun dan dimanapun kita menginginkannya. Tentunya mesin ini didasarkan dengan mesin CNC pada umumnya yang berada di industri dan tentunya harganya terjangkau.

Sistem pengoperasian mesin CNC menggunakan program yang dikontrol langsung oleh komputer. Secara umum konstruksi mesin perkakas CNC dan sistem kerjanya adalah sinkronisasi antara komputer dan mekaniknya. Terdapat 3 motor *Stepper* sebagai penggerak utama dari mesin CNC yang diletakkan di 3 axis yaitu axis X, Y, dan Z. Di sumbu Z terdapat motor *Spindle* yang berguna untuk mengebor atau memotong objek. Pada kenyataannya saat melakukan proses pengeboran atau pemotongan, tingkat kedalaman pahat akan selalu berubah-ubah sesuai dengan proses kerja yang dilakukan sehingga mempengaruhi kecepatan motor *Spindle* dan mengakibatkan tingkat presisi hasil kerja menjadi

berkurang. Proses pengeboran benda kerja memerlukan kecepatan potong yang stabil agar hasil kerja memiliki tingkat presisi yang tinggi.

Agar dapat mewujudkan kebutuhan tersebut, dirancang sebuah mesin CNC *Portable* dengan kontroler PI dan sistem *close loop* pada motor *Spindle* yang mampu membuat respon sistem sesuai dengan yang diinginkan. Pergerakan dari motor *Spindle* dikontrol dengan kontroler PI dengan sistem *close loop* agar saat melakukan pengeboran bisa stabil dan mesin CNC dapat dibawa kemana-mana.

1.2 Perumusan Masalah

Tidak semua industri menggunakan mesin CNC untuk memproduksi sebuah produk, terutama pada industri kecil dan menengah. Karena terbatas nya lahan untuk menempatkan mesin CNC. Karena pada industri sekarang ini, ukuran dari Mesin CNC sangat besar, tergantung dari benda kerja yang dibuat.

Mesin CNC memiliki tingkat kepresisian yang tinggi. agar dapat menghasilkan benda yang halus, kecepatan *Spindle* saat melakukan pemotongan harus dijaga konstan. Kecepatan motor *Spindle* dapat berkurang dikarenakan naiknya torsi beban. Untuk mengatasi naik turunnya kecepatan putar motor DC *Spindle* diperlukan sebuah kontroler yang mampu meminimalkan nilai error kecepatan putar motor DC *Spindle*.

1.3 Batasan Masalah

Mesin CNC *Portable* ini menggunakan 3 Axis, yaitu X,Y,Z dan motor *Spindle* untuk memotong dan mengebor benda kerja. Dimana mesin CNC *Portable* ini berukuran 40x35 cm yang diharapkan bisa melakukan pemotongan objek berukuran 30 x 20 cm. Pada sumbu Z dipasang sebuah motor *Spindle*. Tipe dari motor *Spindle* menggunakan jenis motor DC. Untuk menjaga kestabilan dari motor *Spindle* saat melakukan pemotongan dan pengeboran terhadap benda kerja maka dirancang kontroler PI dengan sistem *close loop*. Sensor yang digunakan merupakan sensor *optocoupler* yang akan membaca kecepatan putar dari motor DC *Spindle*.

1.4 Tujuan

Agar mesin CNC ini cocok diterapkan terutama pada industri kecil dan menengah, maka dibuat bentuk mesin CNC *Portable* dimana fungsinya sama dengan mesin CNC yang lain. Mesin CNC *Portable* ini mudah dipindah tempat dan dibawa kemana-mana.

Mesin CNC *Portable* ini dapat mengontrol atau menjaga kestabilan kecepatan motor *Spindle* saat melakukan proses pemotongan dan pengeboran benda kerja.

1.5 Metodologi Penelitian

Dalam melakukan perancangan controller PI dan mekanik mesin CNC *Portable* terbagi menjadi empat tahap yang meliputi studi literatur, perancangan sistem, uji coba dan hasil pengujian, serta penyusunan laporan.

Pada tahap studi literatur, dilakukan pencarian literatur baik dari buku maupun kumpulan makalah dan jurnal yang mengarah pada topik tugas akhir, kegiatan tersebut dilakukan untuk mencari informasi dan spesifikasi tentang data-data mesin CNC *Portable* yang akan dikontrol motor *Spindle* nya.

Selanjutnya pada perancangan sistem, dibuat program melalui *software* LabVIEW, *software* GRBL dan Arduino untuk mengetahui perhitungan kecepatan motor *Spindle* dan tata letak koordinat motor *Stepper*.

Setelah melalui tahap perancangan sistem, dilakukan uji coba. Pada proses ini, dilakukan uji coba panel untuk mengatur jalannya mesin CNC berdasarkan Gambar yang *diinputkan*.

Tahap terakhir yaitu, penyusunan laporan. Dalam tahap ini, luaran yang diharapkan berupa sebuah laporan yang meliputi semua proses pengerjaan yang dilakukan dalam proses perancangan controller PI pada *Spindle* dan mekanik mesin CNC *Portable*.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Penjelasan mengenai komponen *hardware* maupun *software* pendukung untuk mesin CNC *Portable* dibahas pada bab ini. Penjelasan meliputi jenis mesin CNC, komponen penyusun mesin, cara kerja, dan pengoperasian mesin tersut.

Bab III Perancangan Sistem

Pada bab ini dibahas mengenai perancangan sistem secara keseluruhan. Pembahasan meliputi prosedur koneksi *hardware* dan *software* serta simulasi sistem yang dibuat.

Bab IV Pengujian dan Analisa Data

Hasil dari uji coba program yang dibuat, serta koneksi *hardware* dan simulasi sistem dibahas pada bab ini.

Bab V Penutup

Bab penutup berisi kesimpulan dari hasil pengujian. Selain itu, pada bab ini terdapat kesimpulan serta saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi referensi lanjutan untuk perancangan kontroler PI dan mekanik mesin CNC pada dunia industri secara nyata.

BAB II

KOMPONEN DAN *SOFTWARE* PENDUKUNG MESIN CNC *PORTABLE*

2.1 Mesin CNC^[6]

CNC sendiri merupakan mesin bantu berbasis mekatronika yang diaplikasikan untuk mengatur berbagai macam mesin, seperti mesin perkakas, fabrikasi, perkayuan, ukir, dan lain-lain. Penggunaan mesin ini dalam dunia industri makin pesat perkembangannya, karena ketidakmampuan mesin perkakas manual dalam menghasilkan produk massal dengan hasil konsisten, kuat, berkualitas, dan akurat. Pada dunia industri mesin ini banyak diaplikasikan pada berbagai sektor, mulai dari produksi bagian mesin, kelistrikan, hingga produksi mainan dan perabot rumah tangga. CNC ini merupakan mesin perkakas yang dilengkapi dengan sistem mekanik dan kontrol berbasis komputer yang mampu membaca instruksi kode angka, huruf dan simbol sesuai dengan standar ISO. Adanya mesin CNC berawal dari berkembangnya sistem *Numerically Controlled* (NC) pada akhir tahun 1940-an dan awal tahun 1952-an yang ditemukan oleh John C. Parsons dengan bekerja sama dengan Perusahaan *Servomechanism Massachusetts Institute of Technology MIT*.^[6] Mesin ini dibedakan menjadi 2 jenis berdasarkan jumlah sumbu atau aksis yang dimiliki, yaitu jenis *Lathe/ turning* yang memiliki 2 aksis yaitu X dan Z, serta jenis *Milling* yang mempunyai 3 aksis yaitu X, Y dan Z. Dimana X dan Y berfungsi untuk membentuk kontur horizontal, sedangkan Z menentukan kedalaman potongan mata pahat terhadap benda kerja. Pergerakan pada aksis X, Y, dan Z dilakukan oleh penggerak *Spindle*. Bagian dari mesin ini dibagi menjadi 2 kelompok besar, yaitu mekanik dan elektrik. Pada bagian mekanik, terdiri dari *Spindle*, *Stepper*, dan *Ballscrew*. Sedangkan pada bagian elektrik terdiri dari sensor *limit switch*, kontroler dan *Human Machine Interface* (HMI).

Pengaturan pergerakan aksis pada mesin dilakukan dengan memberikan program ke dalam mesin melalui sebuah PC atau panel. Pembuatan program bisa dilakukan secara langsung pada sebuah PC/panel yang terkoneksi dengan mesin, atau dengan membuat program diluar mesin dengan menggunakan *software* seperti MasterCAM, GRBL, maupun MACH3. Pada bagian axis Z terdapat motor *Spindle* untuk melakukan pengeboran dan pemotongan. Motor yang dipakai berjenis motor DC tipe *high speed* dimana motor ini memiliki kecepatan yang

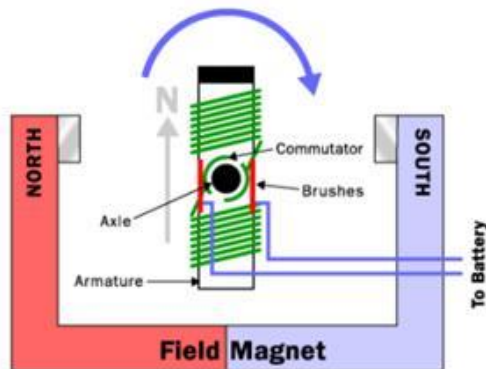
tinggi. Walaupun kecepatannya tinggi, disaat melakukan pekerjaan, motor tersebut nyatanya berkurang kecepatannya dan hal ini bisa mengurangi tingkat kepresisian hasil kerja. Oleh karena itu perlu dirancang suatu kontroler agar kecepatan dari motor DC *Spindle* dapat dikontrol dan perlu perakitan mesin CNC yang rapi agar pergerakan dari mesin tersebut lancar.

2.2 Motor DC^[4]

Motor DC didalam peralatan elektronik telah banyak dijumpai dan digunakan. Motor DC sendiri merupakan jenis motor yang mendapat sumber Direct Current (DC) sebagaimana namanya, menggunakan arus langsung dan tidak langsung/direct-undirectional. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalan torsi yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas, dan selain itu kecepatan motor DC lebih mudah diatur daripada motor AC.

2.2.1 Pengertian Motor DC^[4]

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Bagian utama motor DC adalah stator dan rotor dimana kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Mekanisme kerja motor DC ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Mekanisme Kerja Motor DC

Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak yang disebut komutator. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. Kecepatan putar motor DC (N) dirumuskan dengan persamaan berikut.

$$N = \frac{V_{TM} - I_a R_a}{K \Phi}$$

Keterangan :

V_{TM} : Tegangan Terminal

I_a : Arus Jangkar Motor

R_a : Hambatan Jangkar Motor

K : Konstanta Motor

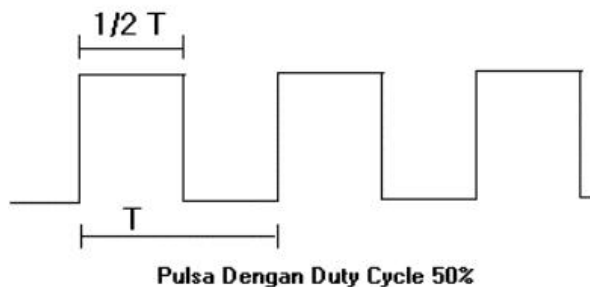
Φ : Fluks Magnet yang terbentuk pada motor

Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor V_{TM} . Metode lain yang biasa digunakan untuk mengendalikan kecepatan motor DC adalah dengan teknik modulasi lebar pulsa atau *Pulse Width Modulation (PWM)*.

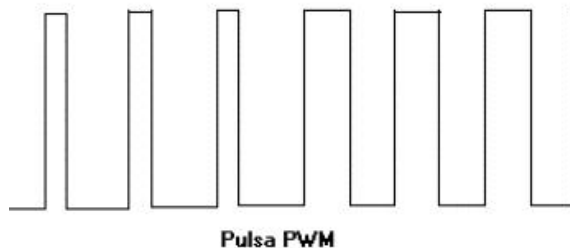
2.2.2 Pengaturan Kecepatan Motor DC^[8]

Cara pengaturan kecepatan yang digunakan adalah dengan menggunakan teknik PWM (*Pulse Width Modulation*). *Pulse Width Modulation* atau PWM adalah teknik untuk memasok tenaga listrik ke beban yang memiliki respon yang relatif lambat. Sinyal suplai terdiri dari rangkaian pulsa tegangan sehingga lebar pulsa individual mengendalikan level tegangan efektif ke beban. Teknik PWM untuk pengaturan kecepatan motor adalah, pengaturan kecepatan motor dengan cara merubah-ubah besarnya *Duty Cycle* pulsa. Pulsa yang berubah ubah

Duty Cycle-nya inilah yang menentukan kecepatan motor. Besarnya amplitudo dan frekuensi pulsa adalah tetap, sedangkan besarnya *Duty Cycle* berubah-ubah sesuai dengan kecepatan yang diinginkan, semakin besar *duty cycle* maka semakin cepat pula kecepatan motor, dan sebaliknya semakin kecil *Duty Cycle* maka semakin pelan pula kecepatan motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah seperti pada Gambar 1, pulsa kotak dengan *Duty Cycle* pulsa 50%. Sebagai contoh bentuk pulsa PWM dapat dilihat pada Gambar 2.3.



Gambar 2.2 Gelombang PWM dengan *Duty Cycle* 50%



Gambar 2.3 Gelombang PWM

Seperti pada Gambar 2.2, semakin besar *Duty Cycle* pulsa kotak, maka semakin lama pula posisi logika *high*. Jika motor diatur agar berjalan ketika diberi logika *high*, maka jika memberi pulsa seperti pada Gambar 2.2 diatas, maka motor akan berada pada kondisi “nyala-mati-nyala-mati” sesuai dengan bentuk pulsa tersesebut. Semakin lama motor berada pada kondisi “nyala” maka semakin cepat pula kecepatan motor tersebut. Motor akan berputar dengan kecepatan maksimum jika

mendapat pulsa dengan *Duty Cycle* 100%. Dengan kata lain motor mendapat logika *high* terus menerus.

Dengan mengatur besarnya *Duty Cycle* pulsa kotak yang dikirimkan, kita dapat mengatur banyaknya logika *high* yang diberikan pada motor, dengan kata lain mengatur lamanya waktu motor untuk berputar dalam satu periode pulsa. Jika lamanya waktu motor untuk berputar dalam satu periode pulsa ini berubah maka kecepatan putaran motor juga akan berubah, sesuai dengan *Duty Cycle* atau waktu motor untuk berputar dalam satu periode pulsa.

2.3 Komponen Penyusun Mesin CNC Portable

Secara garis besar, bagian utama pada mesin CNC dibagi menjadi 3, yang pertama bagian mekanik, lalu bagian elektrik dan yang terakhir adalah bagian kontroler.

2.3.1 Komponen Mekanik

Komponen mekanik merupakan bagian pada mesin yang bergerak, terdiri dari:

a. Motor *Stepper*^[12]

Motor *Stepper* adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor *Stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkannya diperlukan pengendali motor *Stepper* yang membangkitkan pulsa-pulsa periodik. Penggunaan motor *Stepper* memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Dapat dilihat pada Gambar 2.6 tampilan dari motor *Stepper*.

Prinsip kerja motor *Stepper* mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor *Stepper* mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor *Stepper* adalah banyaknya fasa, besarnya nilai derajat per step, besarnya volt tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan.

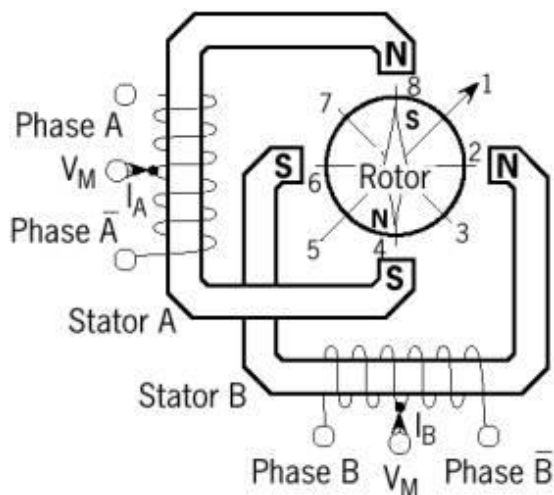
Berdasarkan derajat per stepnya, motor *Stepper* memiliki beberapa jenis spesifikasi masing-masing. Besarnya derajat putaran per step adalah parameter terpenting dalam pemilihan motor *Stepper* karena akan menentukan ukuran langkah gerakan yang paling kecil (resolusi). Tiap-tiap motor *Stepper* mempunyai spesifikasi masing-masing, antara lain 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan

bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 5 prinsip yaitu *Full Step*, *Half Step*, quarter step, eighth step, dan sixteenth step. Dengan *Full Step* berarti motor *Stepper* berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan *Half Step* berarti motor *Stepper* berputar setengah derajat per step dari spesifikasi motor *Stepper* tersebut, quarter step artinya motor tersebut berputar seperempat dari derajat per stepnya, eighth step artinya motor tersebut berputar seperdelapan dari derajat per stepnya, dan sixteenth step artinya motor tersebut berputar seperenambelas dari derajat per stepnya.

Berdasarkan metode perancangan rangkain pengendali motor *Stepper*, motor *Stepper* dapat dibagi menjadi jenis *Unipolar* dan *Bipolar*.

a. *Motor Stepper Unipolar*

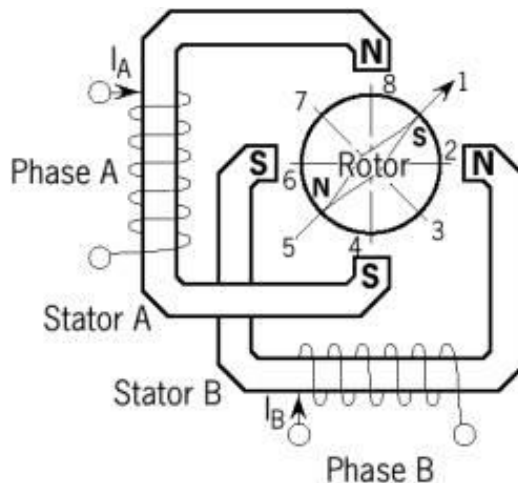
Rangkaian pengendali motor *Stepper Unipolar* lebih mudah dirancang karena hanya memerlukan satu *switch* / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (V_M) pada bagian tengah (*center tap*) dari lilitan, seperti yang terlihat pada Gambar 2.4.



Gambar 2.4 Motor *Stepper Unipolar*

b. *Motor Stepper Bipolar*

Untuk motor *Stepper* dengan lilitan *Bipolar*, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor *Unipolar*. Motor *Stepper Bipolar* memiliki keunggulan dibandingkan dengan motor *Stepper Unipolar* dalam hal torsi yang lebih besar untuk ukuran yang sama, seperti yang terlihat pada Gambar 2.5.



Gambar 2.5 Motor *Stepper Bipolar*



Gambar 2.6 Motor *Stepper*

b. *Motor Spindle*

Motor Spindle merupakan alat yang digunakan untuk melakukan pemotongan maupun pengeboran pada benda kerja. Motor ini ditempatkan pada bagian axis Z dimana pergerakan naik turunnya axis Z digerakkan oleh motor *Stepper*. Terdapat dua jenis motor yang bisa digunakan sebagai motor *Spindle* yaitu motor DC atau motor AC. Masing-masing dari jenis motor tersebut baik motor DC maupun AC memiliki kelebihan dan kekurangan tersendiri. Motor AC memiliki kecepatan yang tinggi dan torsiya juga besar namun untuk pengaturan kecepatannya agak susah. Untuk motor DC kecepatannya tidak secepat motor AC namun untuk jenis *high speed*, motor DC kuat untuk melakukan pemotongan maupun pengeboran dan lebih gampang diatur kecepatannya. Dapat dilihat pada Gambar 2.7 merupakan bentuk motor *Spindle* yang digunakan untuk mesin CNC.



Gambar 2.7 Motor *Spindle*

c. *Ball screw*

Ball screw adalah aktuator linier mekanik yang menerjemahkan gerak rotasi ke gerakan linier dengan sedikit gesekan. Sebuah poros berulir menyediakan jalur untuk bantalan bola yang bertindak sebagai sekrup presisi. Serta mampu menerapkan atau menahan beban dorong tinggi, mereka dapat melakukannya dengan gesekan *internal* minimum. Mereka dibuat untuk menutup toleransi dan karena itu cocok untuk digunakan dalam situasi di mana presisi tinggi diperlukan. Rakitan bola berfungsi sebagai mur sementara poros berulir adalah sekrup. Berbeda dengan timbangan konvensional, *Ball screws* cenderung agak besar, karena perlu ada mekanisme untuk mengedarkan kembali bola. Secara umum bentuk dari *Ball screw* dapat dilihat pada Gambar 2.8.



Gambar 2.8 *Ball screw*

2.3.2 Komponen Elektrik

Komponen elektrik merupakan bagian yang berfungsi memberikan tenaga ke komponen mekanik supaya bergerak sesuai perintah kontroler, yaitu:

a. Arduino^[1]

Arduino adalah pengendali mikro single-*Board* yang bersifat open-source, diturunkan dari *Wiring* platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino.

Arduino Uno menggunakan ATmega328 yang diprogram sebagai USB *to*-serial converter untuk komunikasi serial ke komputer melalui port USB. Adapun data teknis *Board* Arduino Uno seperti pada Tabel 2.1 berikut :

Tabel 0.1 Data Teknis *Board* Arduino Uno

Mikrokontroler	ATMega328
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7-12 V
<i>Input Voltage (limit)</i>	6-20 V
<i>Digital I/O Pins</i>	14 (dimana 6 diantaranya pin <i>output</i> PWM)
<i>Analog Input Pins</i>	6
<i>DC Current per I/O Pin</i>	40 mA

<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	32 KB
SRAM	2 KB
EEPROM	1 KB
<i>Clock Speed</i>	16 MHz

Arduino UNO dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal* (otomatis).

Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jack pusat-positif ukuran 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor POWER.

Kisaran kebutuhan daya yang disarankan untuk *Board* Uno adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v Uno dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *Board* Uno.

Pin listrik adalah sebagai berikut:

- VIN. Tegangan masukan kepada *Board* Arduino ketika itu menggunakan sumber daya *eksternal* (sebagai pengganti dari 5 volt koneksi USB atau sumber daya lainnya).
- 5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya.
- 3v3. Sebuah pasokan 3,3 volt dihasilkan oleh regulator *on-Board*.
- GND. Ground pin.

Selain itu, beberapa pin pada Arduino UNO memiliki fungsi khusus diantaranya :

- Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 *USB-to-Serial* TTL.
- *Eksternal* Interupsi : 2 dan 3. Pin ini dapat dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat *attachInterrupt()* fungsi untuk rincian.

- PWM : 3,5,6,9,10, dan 11. Menyediakan 8-bit *output* PWM dengan *analogWrite()* fungsi.
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan library SPI.
- LED : 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI. LED menyala, ketika pin adalah RENDAH, itu off.

Arduino Uno R3 memiliki 6 *input* analog diberi label A0 sampai A5, masing – masing menyediakan 10-bit resolusi (yaitu 1024 nilai yang berbeda). Secara default sistem mengukur dari ground sampai 5 volt, meskipun mungkin untuk mengubah ujung atas rentang menggunakan pin AREF dan fungsi *analogReference()*. Selain itu, beberapa pin memiliki fungsi khusus :

- TWI : A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- AREF : Referensi tegangan untuk *input* analog. Digunakan dengan *analogReference()*
- RESET

Arduino Uno R3 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada *Board* ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. Firmware '16U2 menggunakan USB *Driver* standar COM, dan tidak ada *Driver eksternal* yang diperlukan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *Board* Arduino. RX dan TX di *Board* LED akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). Fungsi ini digunakan untuk melakukan komunikasi interface pada sistem. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Tampilan dari Arduino UNO dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arduino UNO

b. *Driver Motor*

Driver Motor merupakan alat elektronik yang terdiri dari berbagai IC dan komponen yang menjadi satu yang berfungsi untuk menggerakkan motor DC. Alat ini menerima catu daya dari sumber luar sesuai spesifikasi dari *Driver motor* tersebut. Terdapat pula pin motor pada *Driver motor* yang disambungkan dengan kabel motor dan pin pwm yang tersambung dengan kontroler. Dapat dilihat pada Gambar 2.10 tampilan dari *Driver motor*.



Gambar 2.10 *Driver Motor*

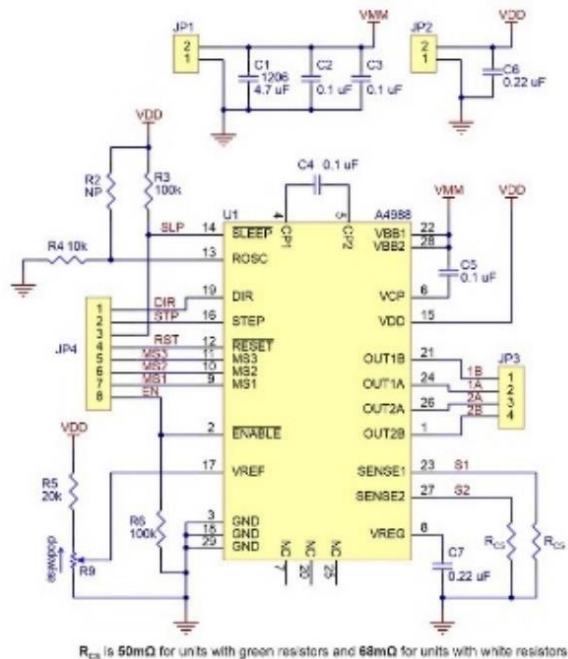
c. *Driver Motor Stepper*^[5]

Salah satu *Driver motor Stepper* yaitu *Driver motor* tipe A4988 yang digunakan untuk menggerakkan motor *Stepper*. *Output* dari *Driver* ini sekitar 2A dan dapat menggerakkan motor *Stepper* dengan pergerakan yang berbeda. Terdapat pin MS untuk pengaturan mode pergerakan dari motor *Stepper*. Jika biasanya pergerakan dari motor *Stepper* sesuai dengan penandaan pada motor *Steppernya* (misalnya 1,8 derajat atau 200 langkah per revolusi) yang berlaku untuk langkah penuh, namun untuk *Driver motor Stepper microstepping* memiliki resolusi yang lebih baik karena dapat mengatur motor *Stepper* sampai 5 jenis pergerakan yang berbeda. Untuk lebih jelasnya mengenai pengaturan logika *state* pada pin MS dapat dilihat pada Tabel 2.2. Sedangkan skematik dari *Driver motor*

A4988 dapat dilihat pada Gambar 2.11 dan tampilan *Driver* A4988 dapat dilihat pada Gambar 2.12.

Tabel 2.2 *Micro Stepping Driver Motor A4988*

MS1	MS2	MS3	Microstep Resolution
<i>Low</i>	<i>Low</i>	<i>Low</i>	<i>Full Step</i>
<i>High</i>	<i>Low</i>	<i>Low</i>	<i>Half Stepper</i>
<i>Low</i>	<i>High</i>	<i>Low</i>	<i>Quarter Step</i>
<i>High</i>	<i>High</i>	<i>Low</i>	<i>Eighth Step</i>
<i>High</i>	<i>High</i>	<i>High</i>	<i>Sixteenth Step</i>



Gambar 2.11 Skematik *Driver Motor A988*



Gambar 2.12 Driver Motor Stepper A4988

d. Sensor Kecepatan^[11]

Sensor Kecepatan disini digunakan untuk mendeteksi kecepatan putar dari motor DC *Spindle*. Sensor kecepatan yang digunakan yaitu sensor *optocoupler* yang dapat dilihat pada Gambar 2.13. Sensor *optocoupler* terdiri dari dua bagian yaitu *transmitter* dan *receiver*. Pada bagian *transmitter* terdapat komponen IR Led yang memancarkan cahaya infra merah. Kemudian cahaya infra merah tersebut diterima untuk *receiver*. *Receiver* yang biasanya digunakan yaitu phototransistor. Prinsip kerja dari sensor ini yaitu jika antara phototransistor dan LED terhalang maka *phototransistor* tersebut akan off sehingga *output* dari kolektor akan berlogika *high*.

Sebaliknya jika antara phototransistor dan LED tidak terhalang maka phototransistor dan LED tidak terhalang maka *phototransistor* tersebut akan on sehingga *output*-nya akan berlogika *low*.



Gambar 2.13 Sensor Kecepatan

c. *Power Supply*

Power Supply adalah perangkat keras berupa kotak yang isinya merupakan kabel-kabel untuk menyalurkan tegangan ke dalam perangkat keras lainnya. Perangkat keras ini biasanya terpasang di bagian belakang (di dalam) casing komputer. *Input Power Supply* berupa arus bolak-balik (AC) sehingga *Power Supply* harus mengubah tegangan AC menjadi DC

(arus searah). Besarnya listrik yang mampu ditangani *Power Supply* ditentukan oleh dayanya dan dihitung dengan satuan Watt.

Power Supply berfungsi sebagai penyuplai tegangan listrik langsung kepada komponen-komponen yang berada di dalam casing komputer. *Power Supply* juga berfungsi untuk mengubah tegangan AC menjadi DC, karena perangkat keras komputer hanya dapat beroperasi dengan arus DC. Tampilan dari *Power Supply* dapat dilihat pada Gambar 2.14.



Gambar 2.14 *Power Supply*

2.3.3 Komponen Kontroler

Komponen kontroler merupakan bagian yang berfungsi mengatur seluruh kegiatan mesin atau merupakan otak dari sebuah mesin CNC. Biasanya kontroler mesin tertanam pada sebuah *operation panel*. Bagian ini berupa sebuah kotak yang berisikan tombol-tombol beserta monitor yang mampu menampilkan segala informasi dan parameter pengaturan mesin CNC. Kotak kontrol ini merupakan bagian langsung yang berhubungan dengan operator.

2.4 CAD/CAM^[10]

Solidworks merupakan sebuah *software* untuk membuat desain sebuah objek yang mampu dikembangkan oleh pengguna untuk melakukan analisis. Dengan *software* ini, pengguna mampu membuat sketsa ide serta bereksperimen dengan berbagai macam desain berbeda untuk membuat model 3D. Pengguna *software* ini beragam, dari berbagai macam kalangan mulai dari pelajar, teknisi, desainer dan kalangan profesional untuk keperluan mendesain *part* kompleks, *assembly* dan menggambar. Mendesain dengan menggunakan Solidworks mempunyai banyak keuntungan, salah satunya lebih menghemat waktu, tenaga dan biaya.

2.5 LabVIEW^[9]

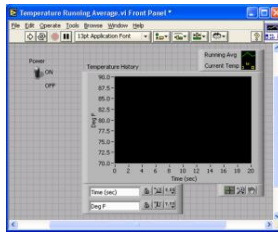
LabVIEW adalah suatu bahasa pemrograman berbasis grafis yang menggunakan icon sebagai ganti bentuk teks untuk menciptakan aplikasi. Berlawanan dengan bahasa pemrograman berbasis text, di mana instruksi menentukan pelaksanaan program, Labview menggunakan pemrograman data *flow*, yang mana alur data menentukan pelaksanaan (*execution*). Tampilan pada Labview menirukan instrument secara virtual.

Dalam LabVIEW, anda membangun antarmuka pemakai dengan satu set peralatan (*Tools*) dan objek-objek. Antarmuka pemakai dikenal sebagai panel depan (*Front Panel*). Anda selanjutnya menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan objek panel muka. *Diagram Block* berisi kode ini. Dalam beberapa hal, *Diagram Block* menyerupai suatu *Flowchart*.

Program LabVIEW disebut sebagai virtual instruments atau VIs sebab operasi dan penampilannya meniru instrumen secara fisik, seperti multimeter dan osiloskop. Labview berisi berbagai macam peralatan untuk menghasilkan ketelitian (*acquiring*), tampilan (*displaying*), dan menyimpan data (*storing data*), seperti halnya perlengkapan untuk membantu anda melakukan pemecahan masalah pengkodean (*code troubleshoot*). Setiap *Virtual Instrumen* (VI) menggunakan fungsi-fungsi yang menggerakkan masukan dari pemakai antarmuka atau sumber lain dan menampilkan informasi itu atau memindahkannya ke file lain atau ke komputer lain. Secara umum LabVIEW memiliki tiga bagian yaitu *Front Panel*, *Diagram Block*, dan Icon dan Connector Pane.

2.5.1 *Front Panel*^[2]

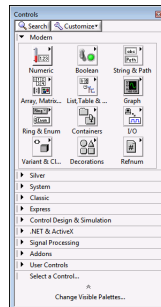
Front Panel merupakan interface antara pengguna (*user*) dengan program. Didalam *Front Panel* terdapat Kontrol (*Input*) dan Indikator (*Output*). Kontrol pada *Front Panel* panel dapat berupa knop, tombol, dial dan lainnya. Sedangkan untuk indikator (*Output*) dapat berupa LED, grafik dan tampilan lainnya. Tampilan pada bagian *Front Panel* dapat dilihat pada Gambar 2.15.



Gambar 2.15 Tampilan *Front Panel* LabVIEW

Kontrol menirukan *input* instrument dan menyuplai data ke *Diagram Block* pada VI yang bersangkutan. Indikator menirukan instrumen keluaran dan menampilkan data yang diperoleh atau dihasilkan oleh *Diagram Block*.

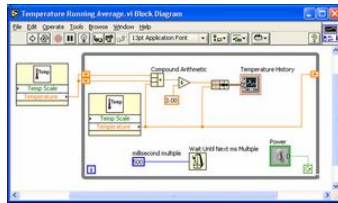
Untuk memasukkan ikon kedalam layer *Front Panel*, kita dapat mengambilnya pada ikon menu bar yang dinamakan *Control Palette*. Dapat dilihat pada Gambar 2.16 *Control Palette* berisi berbagai macam ikon dan *Control Palette* ini hanya dapat digunakan pada layar *Front Panel* saja.



Gambar 2.16 Tampilan *Control Palette* LabVIEW

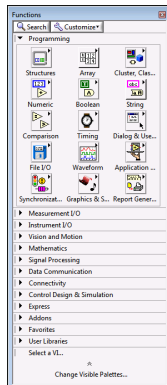
2.5.2 *Diagram Block*^[2]

Setelah membangun panel muka, kita dapat menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan obyek-obyek panel muka. *Diagram Block* berisi *source code* grafis ini. Obyek-obyek panel muka nampak seperti terminal pada *Diagram Block*. *Virtual Instrumen* pada Gambar 2.17 menunjukkan beberapa obyek *Diagram Block* seperti terminal-terminal, fungsi-fungsi, dan alur.



Gambar 2.17 Tampilan *Diagram Block* LabVIEW

Pada *Diagram Block* untuk menambahkan objek pengontrol *Front Panel* dapat mengambil pada *Function Palette*. Dapat dilihat pada Gambar 2.18 *Function Palette* berisi berbagai macam objek pengontrol dan *Function Palette* ini hanya bisa digunakan pada *Diagram Block* Panel.

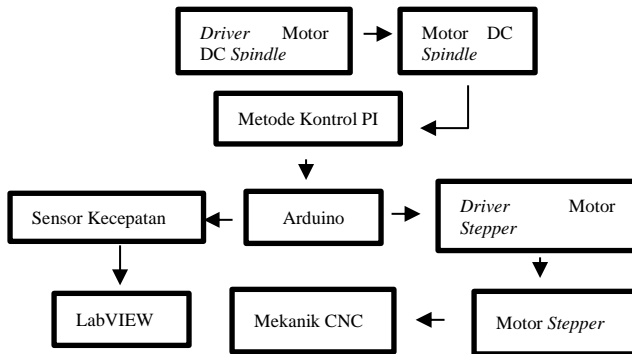


Gambar 2.18 Tampilan *Function Palette* LabVIEW

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

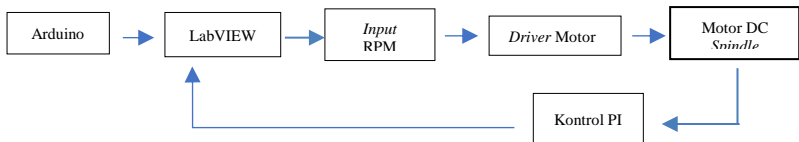
Pada bab ini akan dibahas mengenai Perancangan Pengaturan Kecepatan Motor DC *Spindle* dan Mekanik CNC *Portable* yang meliputi *hardware* dan *software*. Berikut ini adalah blok fungsional sistem secara keseluruhan meliputi cara kerja dalam bentuk alur diagram.



Gambar 3.1 Blok Fungsional Sistem Secara Keseluruhan

Dapat dilihat pada Gambar 3.1, yang nanti akan dibahas selanjutnya yaitu mengenai Pengaturan Kecepatan Motor DC *Spindle* dengan Kontroler PI pada program LabVIEW yang meliputi perancangan *software* dan *hardware* beserta komunikasi Arduino dengan LabVIEW. LabVIEW disini berfungsi sebagai *interface* dari pembacaan sensor kecepatan dan *Set Point* sekaligus dapat mengatur parameter kontroler PI. Selain Pengaturan Kecepatan Motor DC *Spindle*, juga akan dibahas mengenai desain mekanik mesin CNC *Portable* beserta komponen yang digunakan.

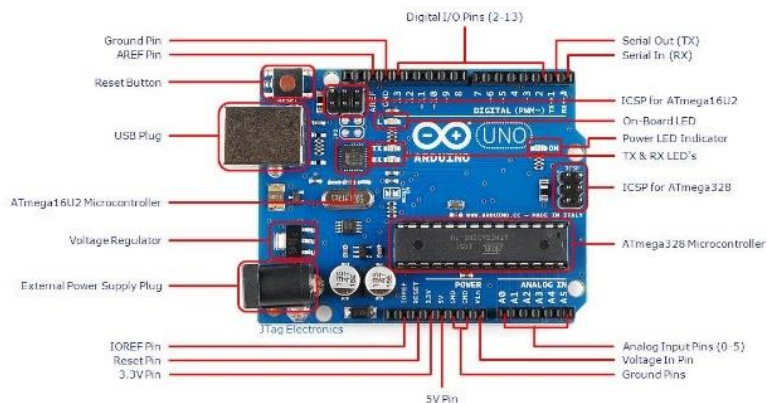
3.1 Perancangan Pengaturan Kecepatan Motor DC *Spindle*



Gambar 3.2 Diagram Alur Perancangan Kendali Motor DC

Alur perancangan motor *Spindle* pada mesin CNC *Portable* yang kami buat dapat dilihat pada Gambar 3.2. Pada bagian awal pengaturan

kecepatan motor DC *Spindle* dengan memprogram Arduino UNO untuk pembacaan sensor kecepatan yang akan ditampilkan di LabVIEW. Agar dapat menjalankan motor DC *Spindle* perlu dikoneksikan antara Arduino UNO dengan LabVIEW dengan menggunakan komunikasi serial pada LabVIEW. Setelah itu maka kita dapat menjalankan program di LabVIEW dan memasukkan nilai *Set Point* motor DC *Spindle*. Ketika motor DC *Spindle* berputar maka otomatis sensor kecepatan membaca kecepatan putar motor DC *Spindle* dan mengirimkan data pembacaan sensor kecepatan ke dalam grafik LabVIEW. Selain grafik pembacaan sensor kecepatan, nilai *Set Point* yang diberikan juga ditampilkan dalam grafik LabVIEW sehingga kita dapat melihat dan membandingkan antara nilai *Set Point* dengan pembacaan sensor kecepatan. Apabila terdapat perbedaan pada nilai pembacaan sensor kecepatan dengan *Set Point* yang diberikan, maka diberikan kontroler PI agar motor DC *Spindle* dapat berputar dengan stabil. Hasil pemberian kontroler PI pada sistem pengaturan kecepatan motor DC *Spindle* selanjutnya ditampilkan pada grafik tersebut.



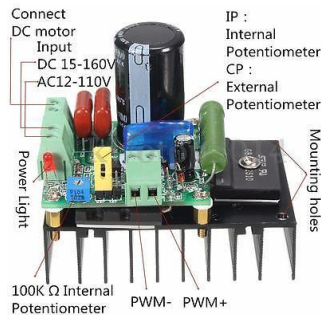
Gambar 3.3 Arduino Uno

Agar dapat menggerakkan motor DC *Spindle* perlu alat penggerak motor DC *Spindle* yang dinamakan *Driver Motor*. *Driver motor Spindle* ini dapat menggerakkan motor DC yang memiliki spesifikasi daya sampai dengan 400 watt. Pada *Driver Motor* ini terdapat pin untuk sumber tegangan dari luar, tegangan yang diterima *Driver Motor* ini bisa tegangan DC maupun AC. Karena tegangan motor yang digunakan merupakan tegangan DC maka sumber tegangan yang digunakan tegangan DC 48

volt. Terdapat pin motor DC yang bisa dihubungkan dengan kabel motor DC *Spindle*. Pada *Driver* ini juga dilengkapi pin PWM *eksternal* maupun *internal*.

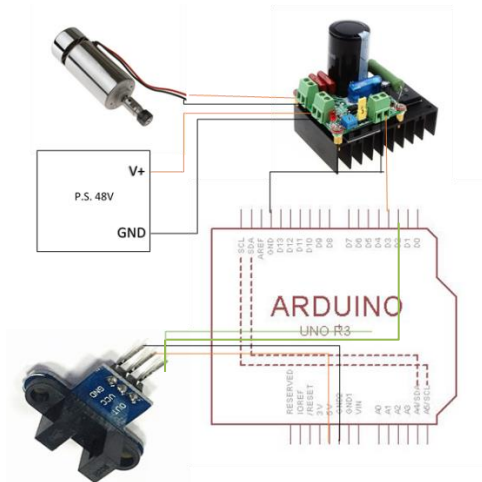
Pada pin PWM *internal* terdapat komponen potensiometer. Potensiometer ini berfungsi untuk mengatur tegangan yang masuk ke motor DC *Spindle* dengan memutar searah jarum jam.

Pada pin PWM *eksternal* dapat disambungkan ke kontroler berupa Arduino Uno yang kemudian bisa deprogram untuk pengaturan nilai PWM nya. Tampilan *Pinout* dari *Driver* motor yang digunakan dapat dilihat pada Gambar 3.4. Sedangkan *Pinout* dari Arduino Uno dapat dilihat pada Gambar 3.3.



Gambar 3.4 *Pinout Driver Motor DC Spindle*

Untuk menggerakkan motor DC *Spindlenya* kami menggunakan pin PWM *eksternal* yang dapat terhubung pada pin PWM Arduino Uno. Terdapat dua pin yang disambungkan yaitu kaki pin PWM positif dan negatif. Untuk kaki pin PWM positif tersambung ke kaki pin digital D3 Arduino Uno dimana kaki pin tersebut berfungsi juga sebagai kaki pin PWM dan untuk kaki pin PWM negatif tersambung ke kaki pin ground GND Arduino Uno. Sedangkan pembacaan kecepatan motor DC *Spindle* dilakukan oleh sensor kecepatan yang tersambung dengan pin D2 Arduino Uno. Sehingga perancangan sistem yang saya buat dapat dilihat pada Gambar 3.5 dan untuk keterangan pin apa saja yang tersambung dapat dilihat pada Tabel 3.1.



Gambar 3.5 Perancangan Pengaturan Kecepatan Motor DC *Spindle*

Tabel 3.1 Keterangan Konfigurasi Pin

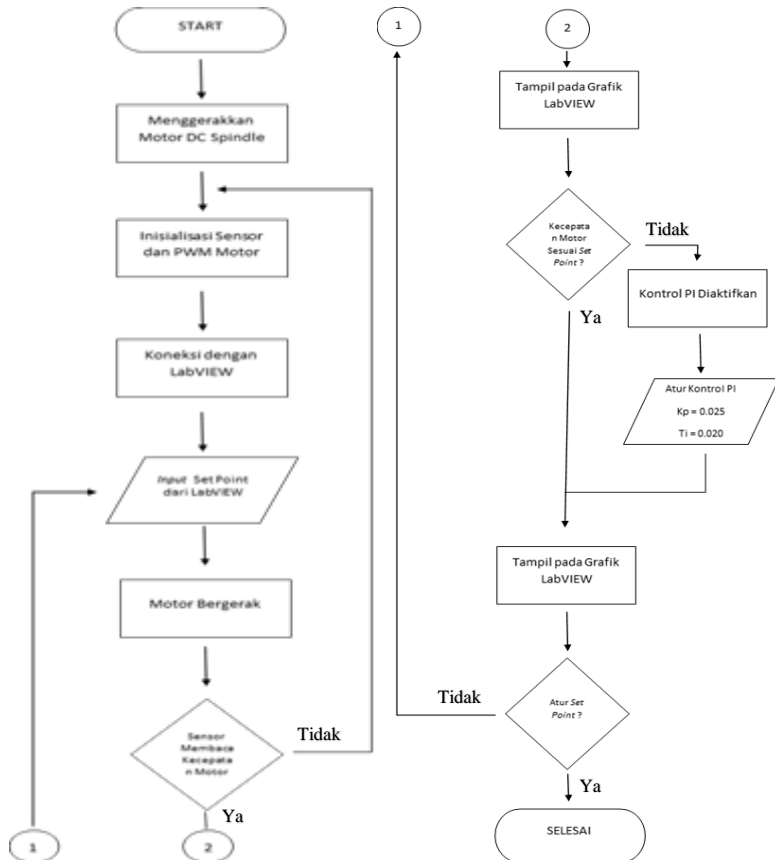
Komponen		Arduino			Power Supply 48V		Motor DC <i>Spindle</i>	
		Data	+5V	GND	V+	V-	VCC	GND
Sensor Kecepatan		D2	VCC	GND				
Driver Motor	PWM	D3	-	GND				
	Sumber				VCC	GND		
	Motor						M+	M-

Pembacaan kecepatan motor DC *Spindle* dalam satuan RPM (*Revolution Per Minute*) dibaca nilainya menggunakan sensor kecepatan *optocoupler*. Pada sensor *optocoupler* ini membaca nilai frekuensi dari piringan yang berlubang tiap perputaran motor DC *Spindle*. Jika dalam satu putaran terdapat satu interupsi, maka sensor mengubah frekuensi motor DC *Spindle* kedalam satuan kecepatan RPM. Didalam perancangan sensor kecepatan yang kami terapkan pada motor DC *Spindle*, kami tempatkan sensor tersebut diatas motor DC *Spindle*. Pada batang poros

motor DC *Spindle* dipasang piringan yang terdapat empat lubang agar saat motor berputar dapat dibaca kecepatannya oleh sensor kecepatan.

Dari pembacaan kecepatan motor DC *Spindle* tersebut kemudian kita dapat merancang kontroler yang sesuai untuk mengontrol kecepatan putar motor DC *Spindle*. Kontroler yang kami gunakan untuk mengatur kecepatan putar motor DC *Spindle* yaitu kontroler PI dimana kontroler ini dapat menstabilkan kecepatan motor.

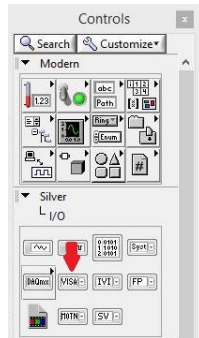
Flowchart sistem yang dikerjakan oleh saya dapat dilihat pada Gambar berikut ini :



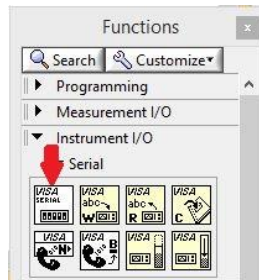
Gambar 3.6 *Flowchart* Sistem

3.2 Perancangan Perangkat Lunak LabVIEW

Pembuatan program untuk menjalankan motor DC *Spindle* pada LabVIEW dimulai dari memasukkan komponen komunikasi serial pada *Diagram Block* LabVIEW yang nantinya tersambung dengan program Arduino. Komunikasi serial yang terdapat pada LabVIEW dinamakan VISA (*Virtual Instrument Software Architecture*) yaitu standard untuk memprogram, mengkonfigurasi, dan memecahkan masalah sistem instrumentasi yang terdiri dari antarmuka, GPIB, VXI, PXI, Serial, Ethernet, dan /atau USB. VISA menyediakan fitur antarmuka antara *hardware* dengan aplikasi LabVIEW. Pada *Diagram Block* yang saya buat, diawal program terdapat VISA *resources name* untuk inisialisasi port COM yang tersambung di komputer kita. VISA *Resources Names* ini dapat dimasukkan pada program LabVIEW melalui palette control pada bagian menu Silver I/O. Kemudian VISA *resources name* dihubungkan dengan VISA *Configure Serial Port* dimana fungsi komponen ini untuk mengatur komunikasi serial antara *hardware* dengan LabVIEW. Pada VISA *Configure Serial Port* terdapat beberapa pin pengaturan komunikasi serial diantaranya pengaturan *baud rate*, data bits, *parity,error*, VISA *resources name out*, *error out*, dan yang lainnya. VISA *Configure Serial Port* terdapat dalam *Function Palette* pada sub menu Serial. Komunikasi serial antara *hardware* dengan LabVIEW harus sama agar dapat menjalankan *hardwarenya* dengan program LabVIEW. Dalam pemrograman di Arduino, untuk dapat menjalankan motor DC *Spindle* menggunakan komunikasi serial dengan *baud rate* bernilai 9600 maka dari itu dalam program LabVIEW diatur nilai *baud ratenya* bernilai 9600. Sedangkan data bits yang digunakan merupakan data bits *default* yaitu bernilai 8. VISA *Resources Names* dapat dicari pada *Control Palette* yang dapat dilihat pada Gambar 3.7, sedangkan VISA *Configure Serial Port* dapat dicari pada *Function Palette* yang dapat dilihat pada Gambar 3.8 dan hubungan keduanya dapat dilihat pada Gambar 3.9.



Gambar 3.7 VISA Resources Names



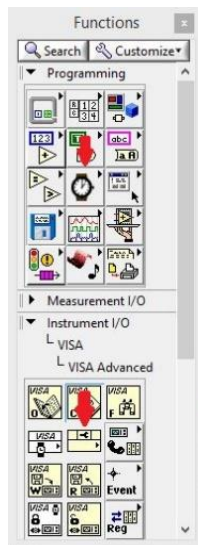
Gambar 3.8 VISA Configure Serial Port



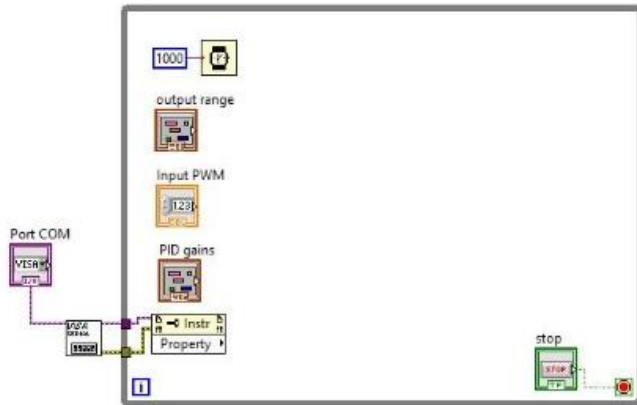
Gambar 3.9 Tampilan Komunikasi VISA pada Diagram Block

Setelah mengatur komunikasi serial LabVIEW maka langkah selanjutnya yaitu membuat tempat dimana semua fungsi dijalankan yang dinamakan *While Loop*. Pada *While Loop* terdapat beberapa komponen masukan dan fungsi diantaranya *output Range* yang digunakan sebagai batasan nilai PWM yang diberikan, *Input PWM* digunakan untuk memberi masukan berupa nilai PWM yang dapat menggerakkan motor

DC *Spindle*, PID *gains* digunakan untuk memberikan nilai dari setiap parameter kontroler PID yaitu K_p , T_i , T_d , dan komponen stop untuk mengentikan program LabVIEW. Selain itu terdapat juga fungsi diantaranya *wait* untuk *delay* programnya, *property node* dihubungkan dengan komunikasi serial VISA. Untuk komponen masukan LabVIEW dapat diambil dari *Control Palette* di *Front Panel* dan untuk blok fungsi dapat diambil dari *Function Palette* di *Diagram Block*. Dapat dilihat pada Gambar 3.10 blok fungsi *While Loop* pada *Function Palette*.



Gambar 3.10 Fungsi pada *While Loop*

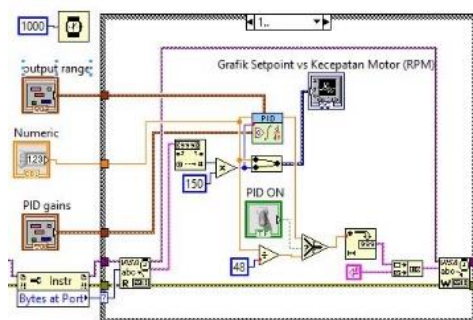


Gambar 3.11 Diagram Block pada Masukan LabVIEW

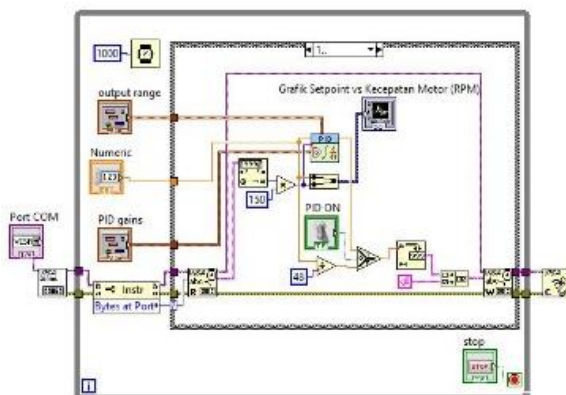
Didalam *While Loop* yang ditunjukkan pada Gambar 3.11 terdapat suatu tempat dimana program tersebut diolah. Tempat ini dinamakan *Structure Case*.

Structure Case ini dapat diambil pada bagian *Programming* kemudian *Structures* di *Function Palette*. Pada *Structures Case* ini terdapat berbagai komponen dan fungsi untuk mengolah masukan yang diberikan agar hasil keluarannya sesuai dengan program yang diberikan Arduino. Lalu komunikasi serial yang telah diatur dihubungkan dengan *VISA Read* agar programnya dapat terbaca pada LabVIEW. Beberapa fungsi diantaranya *Decimal String to Number* untuk mengubah nilai masukan kedalam bilangan bulat. Dari keluaran tersebut dihubungkan dengan perkalian dengan tujuan agar nilai kecepatan motor DC *Spindle* yang keluar sesuai dengan yang ditampilkan pada LabVIEW. Kemudian hasil perkalian tersebut dihubungkan ke fungsi perbandingan. *Input PWM* yang diberikan dibandingkan antara menggunakan kontroler PI dengan tidak menggunakan kontroler PI. Hasil dari perbandingan tersebut kemudian ditampilkan pada *Wave Graphic*. Terdapat komponen *switch* untuk mengatur menggunakan kontroler PID atau tidak. *Input PWM* tadi kemudian dibagi dikarenakan rentang nilai PWM hanya sampai 255 sedangkan nilai kecepatan motor maksimalnya bernilai 12.000, maka dari itu perlu dibagi antara nilai maksimal kecepatan motor dengan nilai PWM dan hasilnya dihubungkan dengan fungsi *select*. Dari keluaran *select* dihubungkan dengan *Number to Decimal String* yang berfungsi mengubah nilai yang sudah diolah tadi kedalam bentuk *String*. Lalu

keluarannya terhubung dengan fungsi *Concatenate String* agar nilainya dapat diterima dan kemudian dibaca oleh *VISA Serial Write*. Antara *VISA resources name out* pada *VISA Serial Read* dengan *VISA resources name* pada *VISA Serial Write* dihubungkan dan keseluruhan program tersebut ditutup dengan menghubungkan pada *VISA Close*. Tampilan *Structures Case* pada program LabVIEW dapat dilihat pada Gambar 3.12 dan keseluruhan program LabVIEW dapat dilihat pada Gambar 3.13.



Gambar 3.12 *Structures Case* Program LabVIEW



Gambar 3.13 Keseluruhan Perancangan Program LabVIEW

Pada layar *Front Panel* LabVIEW terdapat grafik yang berisi dua garis. Garis tersebut merupakan garis masukan nilai PWM dan garis sensor kecepatan. Dengan memberikan masukan berupa sinyal step dan

respon sistem motor DC maka tentunya kontroler yang digunakan tidak diaktifkan dan memberikan tegangan yang diinginkan melalui masukan PWM.

3.3 Perancangan Kontroler PI pada Motor DC *Spindle*

Kontroler PI adalah suatu pengontrol proporsional yang memberikan aksi kontrol proporsional dengan error akan mengakibatkan efek pada pengurangan rise time dan menimbulkan kesalahan keadaan tunak (offset). Suatu pengontrol integral yang memberikan aksi kontrol sebanding dengan jumlah kesalahan akan mengakibatkan efek yang baik dalam mengurangi kesalahan keadaan tunak tetapi dapat mengakibatkan respon transien yang memburuk. Pengetahuan tentang efek yang diakibatkan oleh masing-masing pengontrol tersebut yang nantinya akan digunakan dalam penentuan nilai-nilai penguatan proporsional (K_p) dan integral (K_i).

Penggunaan kontroler PI memiliki tujuan yaitu mempercepat reaksi sebuah sistem dan menghilangkan offset. Hubungan antara penguatan dan efeknya pada kontroler PI dapat dilihat pada Tabel 3.2.

Tabel 3.2 Efek dari Pengontrol P dan I

Penguatan	<i>Rise time</i>	<i>Overshoot</i>	<i>Setling time</i>	<i>Offset</i>
K_p	Menurun	Meningkat	Peubahan kecil	Terjadi
K_i	Menurun	Meningkat	Meningkat	Menghilangkan

Kontroler PI merupakan penggabungan kontroler P dan kontroler I dimana efek dari tiap control digabungkan dan memberikan efek kepada pengontrolan kecepatan motor DC *Spindle*.

3.3.1 Identifikasi *Plant Motor DC Spindle*

Langkah pertama dalam perancangan kontroler PI untuk diterapkan pada motor DC *Spindle* yaitu dengan melihat respon motor dari ketika diberikan masukan pada grafik LabVIEW. Untuk itu perlu dikoneksikan antara Arduino dengan LabVIEW dan diberikan masukan *Set Point* dari LabVIEW. Saat menjalankan motor DC *Spindle* tersebut, *switch* PI dalam keadaan off.

Respon yang dimaksud berupa kecepatan motor DC *Spindle* dan kemudian ditampilkan dalam grafik agar bisa dianalisa. Dari grafik tersebut kita bisa melihat dan merancang kontroler PI yang sesuai agar pergerakan dari motor DC *Spindle* bisa stabil. Kemudian menjalankan motor DC *Spindle* dengan memberikan tegangan masukan berupa nilai

PWM. Untuk memasukkan nilai PWM Arduino menggunakan perintah Serial agar dapat dikendalikan melalui komunikasi serial. Pada Arduino Uno nilai serialnya yaitu 9600 dan mengatur nilai PWM sebagai *output*. Selain perintah memasukkan nilai PWM pada motor DC *Spindle*, didalam program tersebut juga diberikan perintah untuk pembacaan untuk pembacaan nilai kecepatan yang dihasilkan oleh motor DC *Spindle*. Kemudian menjalankan program pembacaan sensor kecepatan dan didownloadkan ke Arduino UNO. Sensor kecepatan akan terus mengcounter dan mengirimkan data ke LabVIEW. Program pembacaan sensor kecepatan pada Arduino dapat dilihat pada Gambar 3.16 dan untuk program keseluruhannya dapat dilihat pada halaman lampiran A.

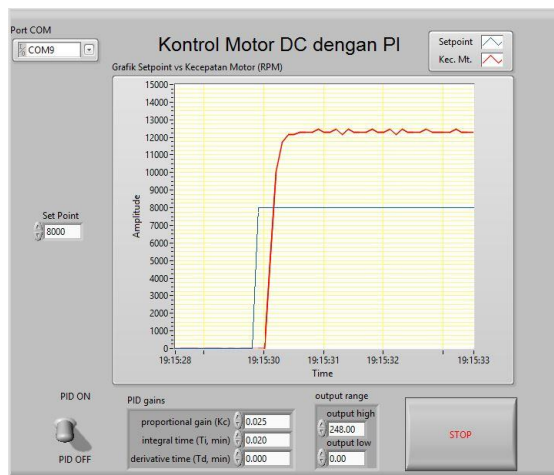
```
void loop(){
  if (millis() - lastmillis == 1000){ //Uptade every one second, t
    detachInterrupt(0); //Disable interrupt when calculating
    rpm = half_revolutions * 15; // Convert frequency to RPM, note:
    Serial.print("RPM =\t"); //print the word "RPM" and tab.
    Serial.println(rpm); // print the rpm value.
    Serial.print("\t Hz=\t"); //print the word "Hz".
    Serial.println(half_revolutions); //print revolutions per second
    half_revolutions = 0; // Restart the RPM counter
    lastmillis = millis(); // Uptade lastmillis
    attachInterrupt(0, rpm_fan, FALLING); //enable interrupt
  }
  // this code will be executed every time the interrupt 0 (pin2)
  void rpm_fan(){
    half_revolutions++;
  }
}
```

Gambar 3.16 Program Sensor Kecepatan pada Arduino

Untuk menjalankan motor DC *Spindle* tersebut saya memakai bantuan *software* untuk LabVIEW. Melalui komunikasi serial yang terdapat pada LabVIEW yang dinamakan VISA, kita dapat mengatur masukan kecepatan motor yang diinginkan. Kemudian pembacaan sensor kecepatan tersebut ditampilkan pada grafik LabVIEW.

Proses pembacaan sensor kecepatan pada motor DC *Spindle* terdapat pada *Diagram Block* LabVIEW dan hasilnya ditampilkan pada grafik LabVIEW. Pada *Diagram Block* program di LabVIEW tersusun oleh beberapa blok function dimana semuanya saling berkaitan mulai dari pembacaan serial LabVIEW dari program Arduino, perhitungan pembacaan sensor kecepatan, dan kemudian menampilkan hasilnya di grafik LabVIEW. Pada perhitungan nilai kecepatan motor DC *Spindle* dilakukan oleh sensor kecepatan dengan memberikan piringan yang berlubang pada motor DC *Spindle*. Satuan nilai kecepatan motor DC *Spindle* yaitu RPM (*Revolution Per Minute*) dimana terdapat beberapa putaran tiap 1 menit tergantung jenis motor DC yang diukur. Sedangkan

program pembacaan sensor kecepatan adalah setiap 100 milidetik. Lalu agar menjadi menit, maka nilai 100 milidetik tersebut dikalikan dengan 60 sehingga menghasilkan 60 detik atau 1 menit. Piringan yang dipasang pada motor DC *Spindle* terdapat 1 lubang sehingga sensor kecepatan membaca pulsa optointerrupter hanya sekali dan nilai yang diberikan yaitu 600 dibagi 1 tetap bernilai 600. Kemudian nilai dari *Set Point* yang diberikan harus dibagi dengan nilai 48. Hal tersebut dikarenakan PWM pada Arduino nilai maksimumnya adalah 248. Padahal *Set Point* maksimum untuk motor DC *Spindlenya* bernilai 12.000. Sehingga agar dapat terbaca kecepatan motor DC *Spindlenya* sampai nilai maksimum maka perlu dibagi antara nilai maksimum *Set Point* dengan nilai PWM yaitu 12.000 dibagi 248 menghasilkan nilai 48.



Gambar 3.17 Grafik Respon Motor DC *Spindle*

Dapat kita lihat pada Gambar 3.17 grafik respon motor DC *Spindle* merupakan grafik respon sistem orde 1. Dikatakan sebagai sistem orde 1 karena grafik yang dihasilkan oleh motor DC *Spindle linear* hasilnya dari beberapa *Set Point* yang diberikan.

3.3.2 Efek Pengontrolan dengan Kontroler PI

Setelah melakukan identifikasi sistem motor DC *Spindle*, kita dapat menentukan *Plant* motor DC tersebut. Diketahui respon sistem motor DC *Spindle* tersebut merupakan sistem orde 1. Pada orde 1 nilai dari *Plant* motor DC *Spindle* dapat dicari dengan rumus

:

$$G(s) = \frac{K}{\tau_s + 1}$$

Dimana : $G(s)$ = *Plant* motor DC
 K = Kecepatan steady motor DC
 τ_s = Waktu transien motor DC

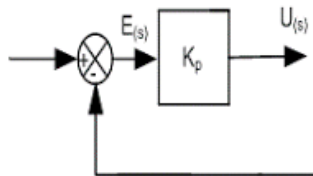
Pencarian parameter kontroler PI disesuaikan dengan model *Plant* yang diperoleh. Perancangan kontroler PI dilakukan untuk setiap kondisi dan dilakukan secara manual. Dengan cara manual maka, untuk setiap parameter kontroler PI dicari dengan memasukkan nilai K_p dan t_i yang sesuai agar pembacaan sensor motor stabil pada *Set Point* yang diberikan. Pengambilan data dilakukan beberapa kali dengan menjalankan motor DC *Spindle* tanpa beban dan dengan diberikan beban.

Pada kontroler P, parameter K_p dapat dicari nilainya sebagai berikut

$$U(t) = K_p e(t)$$

Dimana : $U(t)$ = Sinyal kontrol
 $e(t)$ = Sinyal error
 K_p = Parameter Proportional

Diagram Block untuk kontroler P ini dapat dilihat pada Gambar 3.18



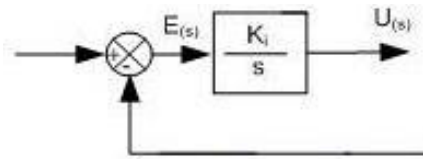
Gambar 3.18 *Diagram Block* Kontroler P

Pada kontroler I, parameter K_i dapat dicari nilainya sebagai berikut:

$$U(t) = K_i \int_0^t e(t) dt, \text{ atau } K_i = \frac{K_p}{t_i}$$

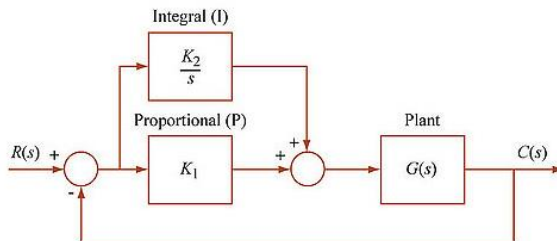
Dimana : $U(t)$ = Sinyal control
 $e(t)$ = Sinyal error
 K_i = Parameter Integral
 K_p = Parameter Proportional

Diagram Block untuk kontroler I ini dapat dilihat pada Gambar 3.19



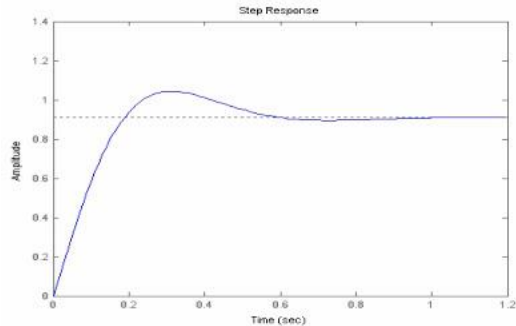
Gambar 3.19 *Diagram Block* Kontroler I

Dengan penggabungan kedua kontroler tersebut menghasilkan suatu kontroler yang dapat memperbaiki nilai error dari sistem yang kita buat. Untuk *Diagram Block* dari kontroler PI dapat dilihat pada Gambar 3.20. Jika dilihat pada Gambar 3.20 kontroler PI terletak setelah perbandingan antara nilai *Set Point* dengan nilai keluaran dari motor DC *Spindle* sehingga ketika terdapat nilai error oleh pembacaan sensor kecepatan maka kontroler PI akan bekerja memperbaiki sistemnya agar nilai error yang dihasilkan bisa diperkecil seminimal mungkin.



Gambar 3.20 *Diagram Block* Kontroler PI

Dari Gambar 3.20 kita bisa tahu bahwa nilai dari kontroler PI memberikan pengaruh pengontrolan kecepatan motor. Setiap data yang berbeda memiliki parameter kontrol yang berbeda pula. Maka untuk itu dari setiap nilai parameter yang diperoleh dirata-rata nilainya sehingga menghasilkan nilai yang sesuai untuk diterapkan pada kecepatan motor DC *Spindle*. Setelah parameter diberikan, akan memberikan sinyal kontrol untuk mengontrol *Plant* dari motor DC *Spindle* agar nilai error yang dihasilkan dapat diminimalisir. Sehingga respon yang dihasilkan pada sistem dapat dilihat pada Gambar 3.21.



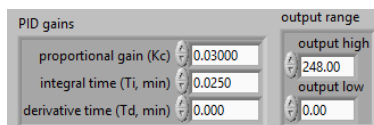
Gambar 3.21 Respon Sistem Terhadap Aksi Kontroler PI

3.3.3 Trial and Error Kontroler PI

Setelah melihat dan menganalisa parameter yang cocok dengan respon motor, maka selanjutnya dapat merancang kontroler PI yang difungsikan pada pengaturan kecepatan motor DC *Spindle*.

Sesuai dengan *Diagram Block*, maka nilai dari hasil tersebut dimasukkan pada bagian pengontrolan setelah menerima *feedback* dari sensor optocoupler. Kontroler PI tersebut memberikan sinyal kontrol untuk diterapkan pada sistem pengaturan kecepatan motor DC *Spindle*.

Parameter yang didapat dari hasil perhitungan kemudian dimasukkan pada program LabVIEW. Pada program LabVIEW terdapat fungsi pengaturan kontrol PID secara otomatis sehingga nilai dari parameter yang dicari tadi dapat dijalankan dengan otomatis menstabilkan kecepatan motor DC *Spindle*. Parameter yang dimasukkan pada program LabVIEW berupa nilai K_p dan τ_i sedangkan nilai τ_d dibiarkan bernilai 0.



Gambar 3.22 Program Kontrol PI di Arduino

Pemrograman pada LabVIEW dilakukan setelah mendapatkan data dari parameter K_p dan τ_i dan memasukkan nilai tersebut kedalam box dari setiap parameter seperti yang ditunjukkan pada Gambar 3.22. Dalam program tersebut tidak hanya terdapat fungsi PID tetapi juga terdapat program pembacaan kecepatan motor DC *Spindle* oleh sensor

optocoupler. Terdapat rangkaian *switch* untuk menentukan penggunaan kontroler PI atau tidak. Cara kerja dari program tersebut apabila kecepatan motor DC *Spindle* dibawah nilai stabil maka fungsi PID akan bekerja secara otomatis menstabilkan kecepatan motor DC *Spindle*. Proses penstabilan kecepatan dengan mengirim *feedback* ke motor DC *Spindle* dengan metode PWM. Untuk keseluruhan program kontrol PI pada Arduino dapat dilihat pada lampiran.

3.4 Mekanik Mesin CNC

Secara umum mesin CNC terbagi menjadi dua bagian, bagian elektrik dan mekanik. Pada bagian mekanik ini terdiri dari berbagai macam komponen penggerak mesin yang disatukan sehingga satu sama lain bergerak sesuai dengan yang kita inginkan. Didalam pembuatan mekanik mesin CNC *Portable* ini perlu perancangan yang rapi dan akurat karena hal ini berpengaruh terhadap tingkat kepresisian mesin CNC *Portable*.

3.4.1 Komponen Pembentuk Mesin CNC

Dalam perakitannya tentunya terdapat komponen-komponen penggerak mesin CNC diantaranya motor *Stepper* dan motor DC *Spindle*. Agar mesin CNC *Portable* ini dapat mengebor dan mengukir kayu dengan baik maka keseluruhan komponen penggerak mesin CNC *Portable* harus berkesinambungan dan gerakannya sesuai dengan perintah yang diberikan.

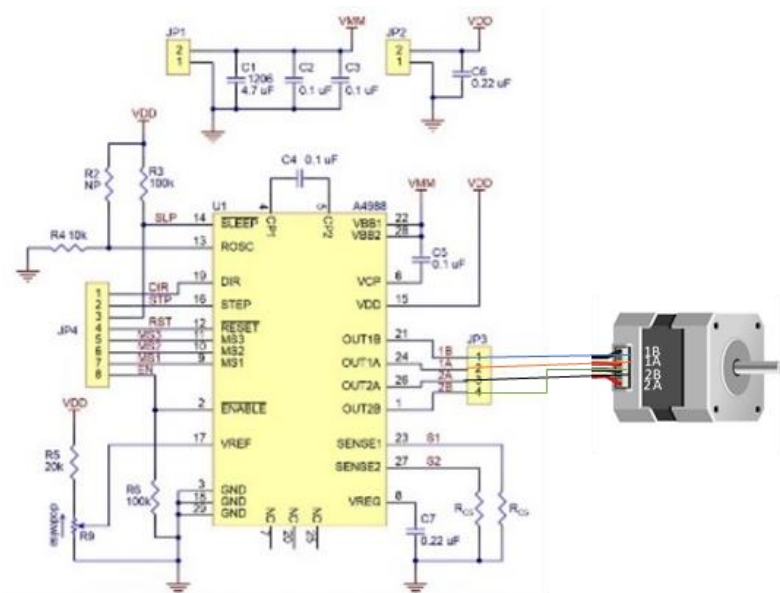
a. Motor Stepper

Motor *Stepper* pada mesin CNC *Portable* merupakan komponen aktif yang bisa menggerakkan mesin. Terdapat 4 motor *Stepper* yang digunakan, 2 motor *Stepper* ditempatkan sumbu Y, 1 motor *Stepper* ditempatkan pada sumbu X, dan pada sumbu X terdapat juga sumbu Z sebagai penggerak dari motor DC *Spindle*. Pada sumbu X dan Y dipasang belt motor agar mesin dapat bergerak sedangkan untuk sumbu Z menggunakan *Ballscrew* untuk menggerakkan motor DC *Spindle*.

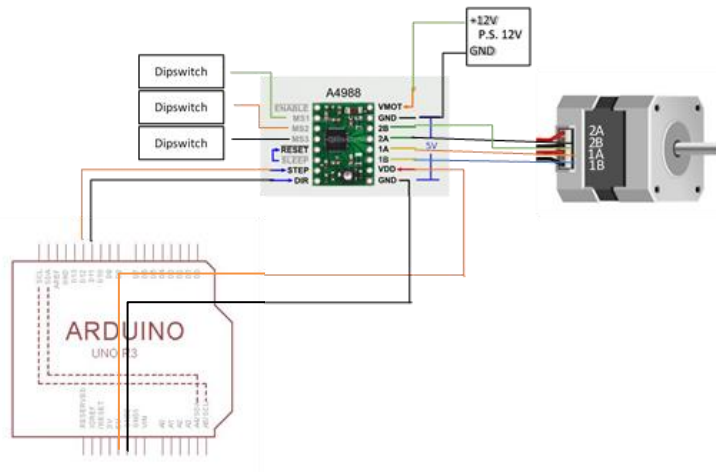
Motor *Stepper* ini agar dapat bergerak perlu disambungkan dengan *Drivernya*. *Driver* motor yang digunakan yaitu *Driver* motor a4988. Pin konfigurasi dari *Driver* motor dengan motor DC *Spindle* dapat dilihat pada Gambar 3.23 dan keterangan pin yang tersambung dapat dilihat pada Tabel 3.3. Sehingga perancangan Arduino dengan motor *Stepper* dapat dilihat pada Gambar 3.24.

Tabel 3.3 Pin Konfigurasi *Driver Motor A4988*

No.	Driver Motor	Pasangan
1.	VMOT	(V+) Motor <i>Power Supply</i> (8-35V)
2.	GND	(GND) Motor <i>Power Supply</i> (8-35V)
3.	2B	2B Motor <i>Stepper</i>
4.	2A	2A Motor <i>Stepper</i>
5.	1A	1A Motor <i>Stepper</i>
6.	1B	1B Motor <i>Stepper</i>
7.	VDD	VCC
8.	GND	GND
9.	DIR	Pin Digital Arduino
10.	STEP	Pin Digital Arduino
11.	SLEEP	RESET
12.	RESET	RESET
13.	MS3	DIPSWITCH
14.	MS2	DIPSWITCH
15.	MS1	DIPSWITCH
16.	ENABLE	VCC



Gambar 3.23 Wiring Diagram *Driver Motor A4988* dengan Motor *Stepper*



Gambar 3.24 Wiring Diagram Driver Motor A4988

b. Motor DC *Spindle*

Motor DC *Spindle* pada CNC *Portable* ini digunakan untuk melakukan *milling* pada benda kerja. Motor DC *Spindle* diletakkan pada axis Z yang terpasang pada axis X. Pada bagian motor DC *Spindle* terdapat piringan yang terletak dibagian atas yang digunakan untuk pembacaan sensor kecepatan. Piringan tersebut memiliki empat lubang bertujuan agar sensor dapat membaca kecepatan setiap kali motor *Spindle* berputar. Data teknis motor DC *Spindle* dapat dilihat pada Tabel 3.3

Tabel 3.4 Data Teknis Motor DC *Spindle*

<i>Spindle</i> Motor	Brushless DC Motor
Tegangan	12-48 V DC
Daya	300 Watt
Kecepatan	3000-12000r/min (12 V-3000 putaran 24 V-6000 putaran 36 V-9000 putaran 48 V-12000 putaran)
Torsi	400mN.m
Resistansi Insulalasi	>2 megaohms
Kekuatan <i>Dielectric</i>	400V

Diameter	52mm
Panjang collet	35mm
Diameter collet	16mm
Panjang Motor	175mm (termasuk penjepit motor dan motor)
<i>Spindle radial runout</i>	Rentang 0.01-0.04

Kabel *power* dari motor DC *Spindle* dihubungkan dengan *Driver* motor sebagai media yang menyalurkan tegangan dari *Power Supply*. Lalu *Driver* motor tersebut memiliki *port pwm eksternal* yang dihubungkan dengan Arduino. Berikut ini adalah data teknis dari *Driver* motor DC *Spindle* yang ditunjukkan oleh tabel 3.5.

Tabel 3.5 Data Teknis *Driver* Motor DC *Spindle*

No.	Fitur
1.	Bisa mendapat masukan sumber AC atau DC. Tidak masalah dengan polaritas sumber listrik.
2.	Ruang lingkup tegangan tinggi . Mendukung sumber DC 12-160 V dan sumber AC 9-110 V.
3.	Radiator besar dengan lubang pemasangan.
4.	Memiliki daya kontrol yang tinggi (maksimal 10A).
5.	Memiliki fungsi PWM eksternal dengan syarat 3,5-12 V VPP dan frekuensi 1K-10KHz
6.	Dilengkapi dengan proteksi sekering dan performa yang lebih baik.
7.	Memiliki daya 400 watt dilengkapi indikator dan lebih nyaman digunakan.

c. Kerangka Mesin CNC *Portable*

Mesin CNC *Portable* dibuat dengan menggunakan bahan metal dan akrilik yang ditunjukkan pada Gambar 3.25. Penggunaan bahan metal dan akrilik tersebut bertujuan membuat mesin CNC *Portable* lebih kokoh dan juga tidak terlalu berat. Pada bagian meja kerja mesin CNC *Portable* menggunakan bahan kayu dan di sisi kanan kiri dipasang bahan metal untuk menyangga mesin CNC *Portable*. Di bagian meja kerja tersebut juga merupakan tempat untuk benda kerja yang akan dikerjakan oleh mesin CNC *Portable* sehingga penggunaan bahan metal disamping kanan

kiri meja kerja agar tidak menimbulkan getaran berlebih yang bisa mengganggu kinerja mesin *CNC Portable*. Tampilan dari meja kerja *CNC Portable* dapat dilihat pada Gambar 3.25.



Gambar 3.25 Meja Kerja Mesin *CNC Portable*

Pada bagian samping kanan dan kiri terpasang motor *Stepper* dengan diberikan gear dan fanbelt untuk menggerakkan mesin *CNC Portable*. Tidak hanya bagian samping kanan dan kiri, tapi juga bagian belakang dan atas yang juga terpasang motor *Stepper* dengan gear dan fanbelt. Perbedaannya pada bagian sumbu Z terdapat *Holder* untuk menopang motor DC *Spindle* yang menggunakan *Ballscrew* untuk pergerakannya.

Holder pada sumbu Z sebagai penopang motor *Stepper* dan motor *Spindle* dibuat dengan menggunakan bahan akrilik. Akrilik tersebut memiliki ketebalan 5mm agar akrilik tersebut dapat menopang dengan kuat komponen yang terpasang pada *Holder* sumbu Z seperti motor *Stepper*, motor DC *Spindle*, dan *Ballscrew*. Dapat dilihat pada Gambar 3.26 tampilan dari *Holder* pada sumbu Z.

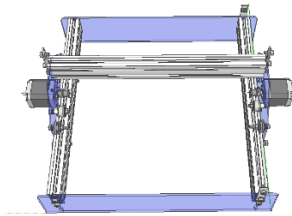


Gambar 3.26 *Holder* Sumbu Z

3.4.2 Dimensi dari Mesin CNC *Portable*

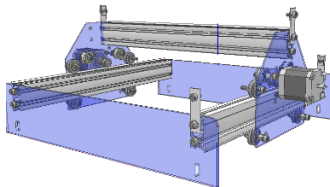
Pada dimensi mesin CNC *Portable* yang dibuat, desain mesin dibagi menjadi 4 bagian utama. Hal ini dilakukan agar tiap bagian mesin dapat diprogram dan bergerak seperti benda aslinya. 4 bagian utama, yaitu:

1. Bagian pertama yaitu meja benda, dapat dilihat pada Gambar 3.27. Bagian ini merupakan *base part* yang berfungsi untuk menempatkan benda kerja saat mesin beroperasi.



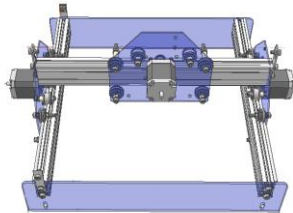
Gambar 3.27 Desain Meja Kerja pada Mesin CNC *Portable*

2. Yang kedua merupakan *Column* atau yang bisa berfungsi sebagai aksis Y pada mesin CNC Portbale. Desain dari bagian ini tertera pada Gambar 3.28. Bagian mesin ini dapat bergerak maju mundur sesuai dengan perintah yang diberikan operator



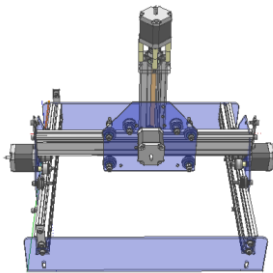
Gambar 3.28 Desain *Column* Aksis Y

3. Bagian ketiga, pada mesin disebut *Spindle Carriage*. Pada simulasi, bagian ini bisa berfungsi sebagai aksis X pada mesin. Desain benda yang dibuat pada simulasi dapat dilihat pada pada Gambar 3.29. Bagian mesin ini bergerak sesuai dengan aksis X, yaitu kiri dan kanan.



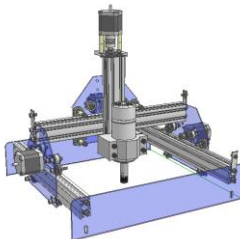
Gambar 3.29 Desain *Spindle Carriage* Aksis X

4. Yang terakhir adalah bagian *Tool Holder*, bagian ini berfungsi sebagai aksis Z. Desain dari bagian ini dapat dilihat pada Gambar 3.30. Seperti pada umumnya aksis Z mesin cnc, bagian ini bergerak naik turun sesuai perintah yang diberikan operator melalui program *G-code*.



Gambar 3.30 Desain *Tool Holder* Aksis Z

5. Setelah itu, dilakukan perangkaian masing-masing *part* hingga membentuk sebuah mesin CNC *Milling* yang utuh seperti pada Gambar 3.31



Gambar 3.31 Desain Mesin CNC *Portable*

3.4.3 Tata Letak Komponen pada Mesin CNC *Portable*

Dalam perakitan mesin CNC *Portable* terdapat berbagai komponen penggerak mesin, seperti motor *Stepper*, *Ballscrew*, belt, dan motor DC *Spindle*. Peletakan komponen tersebut didalam kerangka mesin CNC perlu sesuai ukuran dan tempat yang ditentukan.

Selain peletakan komponen pada kerangka mesin CNC *Portable*, perlu juga memperhatikan bahan pembuat mesin. Bahan yang digunakan harus kuat dan kokoh agar saat mesin bekerja tidak terdapat getaran berlebihan yang. Hal tersebut agar hasil kerja mesin CNC *Portable* bisa presisi dan rapi sehingga sesuai dengan Gambaran yang kita inginkan.

Pergerakan pada setiap sumbu X,Y, dan Z ini dilakukan oleh motor *Stepper*. Motor *Stepper* tersebut dipasang gear dan terdapat gear tambahan untuk dipasang belt agar tiap sumbu dapat bergerak dengan baik. Pada sumbu Z dipasang komponen *Ballscrew* agar sumbu Z dapat bergerak naik turun dengan baik. Supaya *Ballscrew* bisa bergerak naik turun, motor *Stepper* dipasang gear yang dikopel dengan gear yang tersambung dengan *Ballscrew* menggunakan belt. Untuk tampilan tata letak motor *Stepper* dapat dilihat pada Gambar 3.32, Gambar 3.33, Gambar 3.34, dan Gambar 3.35. Sedangkan tampilan tata letak motor DC *Spindle* dapat dilihat pada Gambar 3.36. Untuk perancangan keseluruhan mesin CNC *Portable* dapat dilihat pada bagian Lampiran C.



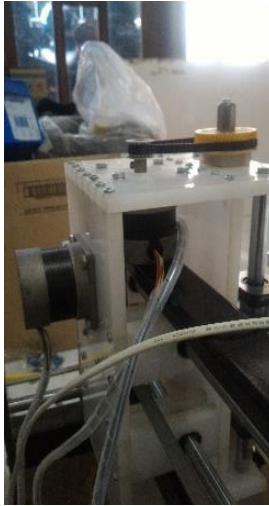
Gambar 3.32 Motor *Stepper* Sumbu Y Tampak Samping Kanan



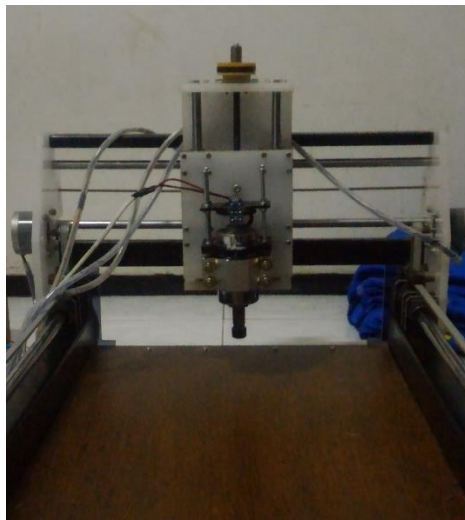
Gambar 3.33 Motor *Stepper* Sumbu Y Tampak Samping Kiri



Gambar 3.34 Motor *Stepper* Sumbu X Tampak dari Belakang



Gambar 3.35 Motor *Stepper* Sumbu Z Tampak dari Tengah



Gambar 3.36 Motor DC *Spindle* pada Sumbu Z

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISA DATA

Dalam mengerjakan alat TA ini beberapa data diperoleh dari hasil pengujian pada alat TA kami terkait dengan pengaturan kecepatan motor DC *Spindle* dengan kontroler PI dan pergerakan dari motor *Stepper* pada mesin CNC *Portable*. Dari hasil pengujian tersebut didapat data yang dibandingkan apakah sudah sesuai dengan perancangan yang dibuat.

4.1 Pengujian Pergerakan Motor *Stepper*

Pada pengujian pergerakan motor *Stepper* tersebut digerakkan secara manual olah tombol yang berada pada *interface* Mesin CNC *Portable*. Pemberian sinyal oleh *interface* melalui kontroler Arduino Uno pada bagian sumbu yang ingin digerakkan. Dalam pergerakan motor *Stepper* tersebut dapat diatur menggunakan *Full Step* atau *Half Step*. Pengaturan pergerakan tersebut menggunakan pin MS1, MS2, dan MS3 pada *Driver* motor A4988 dengan mengatur logika *state* dari pin tersebut. Untuk melihat sejauh mana perpindahan yang terjadi maka dihitung jarak perpindahan motor *Stepper* tersebut dalam satuan derajat. Berikut adalah data yang diperoleh :

Tabel 4.1 Data *Full Step*

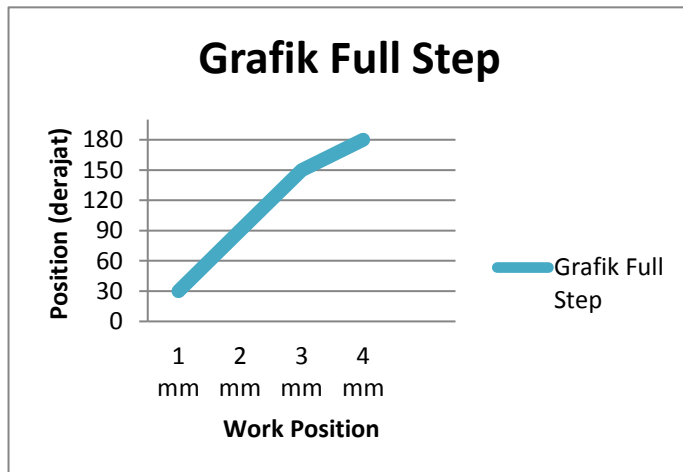
No	M1	M2	M3	<i>Machine Position</i>	<i>Work Position</i>	Perpindahan (derajat)
1	0	0	0	1 mm	1 mm	30
2	0	0	0	2 mm	2 mm	90
3	0	0	0	3 mm	3 mm	150
4	0	0	0	4 mm	4 mm	180

Tabel 4.2 Data *Half Step*

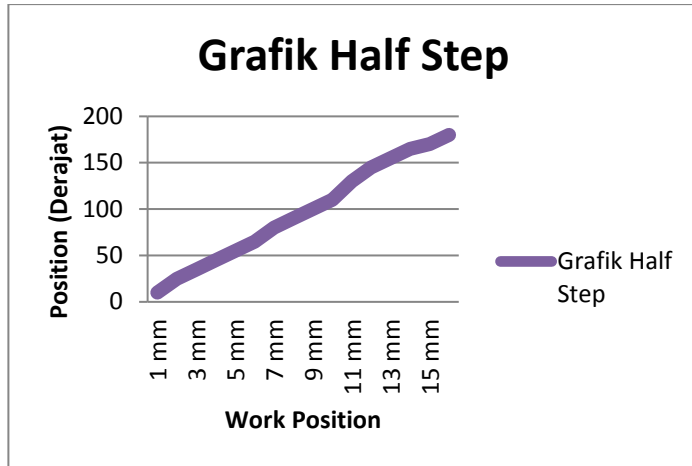
No	M1	M2	M3	<i>Machine Position</i>	<i>Work Position</i>	Perpindahan (derajat)
1	1	0	0	1 mm	1 mm	10
2	1	0	0	2 mm	2 mm	25
3	1	0	0	3 mm	3 mm	35
4	1	0	0	4 mm	4 mm	45
5	1	0	0	5 mm	5 mm	55
6	1	0	0	6 mm	6 mm	65
7	1	0	0	7 mm	7 mm	80
8	1	0	0	8 mm	8 mm	90
9	1	0	0	9 mm	9 mm	100
10	1	0	0	10 mm	10 mm	110

No	M1	M2	M3	Machine Position	Work Position	Perpindahan (derajat)
11	1	0	0	11 mm	11 mm	130
12	1	0	0	12 mm	12 mm	145
13	1	0	0	13 mm	13 mm	155
14	1	0	0	14 mm	14 mm	165
15	1	0	0	15 mm	15 mm	170
16	1	0	0	16 mm	16 mm	180

Data tersebut kemudian diplot dalam bentuk grafik untuk mengetahui perpindahan sudutnya. Gambar 4.1 merupakan data *Full Step* dari Tabel 4.1, sedangkan Gambar 4.2 merupakan data *Half Step* dari Tabel 4.2.



Gambar 4.1 Grafik *Full Step*



Gambar 4.2 Grafik *Half Step*

Dari grafik yang ditunjukkan oleh Gambar dan Gambar dapat dilihat bahwa pada metode *Half Step* perputarannya lebih halus dikarenakan pada metode *Half Step* memiliki sudut pergerakan (derajat) yang lebih kecil dibandingkan dengan metode *Full Step*. Pada grafik tersebut dapat dilihat bahwa pada metode *Full Step* untuk mencapai 180 derajat hanya membutuhkan 4 langkah sedangkan pada metode *Half Step* untuk mencapai 180 derajat membutuhkan 16 langkah.

4.2 Pengujian Kecepatan Motor pada LabVIEW

Pada pengujian ini dilakukan dengan cara memberikan masukan berupa tegangan dari *Power Supply* yang terhubung dengan *Driver* motor DC *Spindle* untuk menggerakkan motornya. Pengambilan data dilakukan secara manual dengan melihat respon dari motor DC *Spindle* pada grafik LabVIEW. Data yang diambil dengan kecepatan 10.000 rpm sampai 12.000 rpm dikarenakan perlu kecepatan yang tinggi untuk melakukan pengeboran sehingga nilai kecepatan 10.000 rpm keatas akan dikontrol menggunakan kontroler PI agar stabil. Tidak hanya pengontrolan PI saja yang diuji, akan tetapi akan dibandingkan pula dengan menggunakan kontroler PID pada pengontrolan kecepatan motor DC *Spindle*.

Selain itu tegangan dari motor DC *Spindle* terdapat perbedaan antara sebelum diberikan pengontrolan dan sesudah diberikan pengontrolan. Pada Tabel 4.3 diperlihatkan tegangan motor sebelum diberikan pengontrolan PI kemudian ditampilkan dalam bentuk grafik

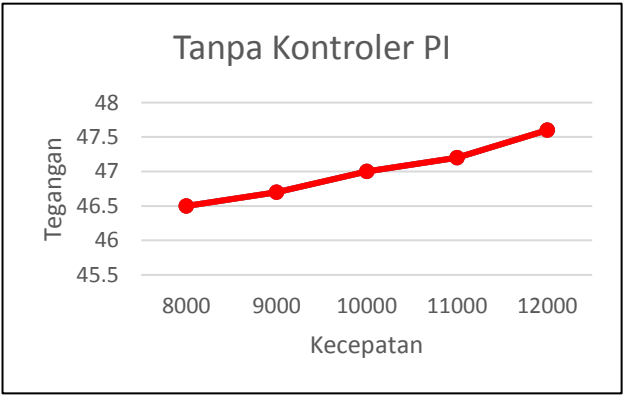
pada Gambar 4.3 dan pada Tabel 4.4 diperlihatkan tegangan motor sesudah diberikan pengontrolan PI kemudian ditampilkan dalam bentuk grafik pada Gambar 4.4.

Tabel 4.3 Tegangan Motor Sebelum Pengontrolan PI

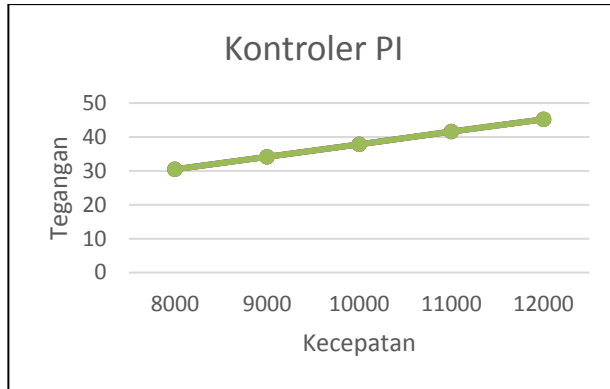
<i>Set Point</i>	<i>Tegangan Motor DC Spindle</i>
8000 RPM	46.5V
9000 RPM	46.7V
10000 RPM	47.0V
11000 RPM	47.2V
12000 RPM	47.6V

Tabel 4.4 Tegangan Motor Sesudah Pengontrolan PI

<i>Set Point</i>	<i>Tegangan Motor DC Spindle</i>
8000 RPM	30.47V
9000 RPM	34.12V
10000 RPM	37.8V
11000 RPM	41.6V
12000 RPM	45.2V



Gambar 4.3 Tegangan dan Kecepatan Tanpa Kontroler PI



Gambar 4.4 Tegangan dan Kecepatan dengan Kontroler PI

Dari data yang diketahui bahwa terdapat perbedaan tegangan antara sebelum diberikan kontrol PI dan sesudah diberikan kontrol PI dikarenakan kontrol PI berfungsi otomatis menyetabilkan kecepatan motor DC *Spindle* sesuai dengan *Set Point*. Saat sebelum diberikan kontrol PI, kecepatan motor DC *Spindle* tidak sesuai dengan stabil dan tegangan yang dihasilkan juga berbeda dan kemudian diberikan kontroler PI maka kecepatan dan tegangan motor DC *Spindle* turun menyesuaikan dengan nilai dari *Set Point*.

4.2.1 Pengujian Pengaturan Kecepatan Motor Tanpa Kontroler

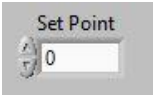
Pada pengujian ini akan diambil beberapa data dengan nilai *Set Point* antara 10.000 sampai 12.000 rpm. Pada kecepatan tersebut motor DC *Spindle* dapat melakukan pengeboran dengan baik. Dari setiap nilai *Set Point* yang diberikan tidak akan diberikan kontroler terlebih dahulu untuk melihat respon motor DC *Spindle* yang dihasilkan pada grafik LabVIEW.

Langkah pertama yang harus dilakukan untuk mengambil data pada pengujian ini adalah dengan mendownloadkan program pembacaan sensor kecepatan pada Arduino. Kemudian mengkomunikasikan Arduino dengan LabVIEW melalui komunikasi serial VISA LabVIEW pada layer *Front Panel*. Untuk tampilan *port* serial pada LabVIEW dapat dilihat pada Gambar 4.5.



Gambar 4.5 Komunikasi Serial VISA LabVIEW

Setelah itu memberikan nilai *Set Point* pada motor DC *Spindle* melalui blok fungsi *numeric control* LabVIEW. Nilai *Set Point* tersebut akan menggerakkan motor DC *Spindle* melalui *Driver* motor DC *Spindle* yang terhubung dengan Arduino melalui fungsi PWM yang terdapat pada Arduino. Oleh sebab itu pada penginputan nilai *Set Point* harus dibagi dengan *Range* PWM sehingga menghasilkan nilai *Set Point* yang sesuai. Tampilan dari blok fungsi *numeric control* dapat dilihat pada Gambar 4.6 dan *Range* PWM dapat dilihat pada Gambar 4.7.

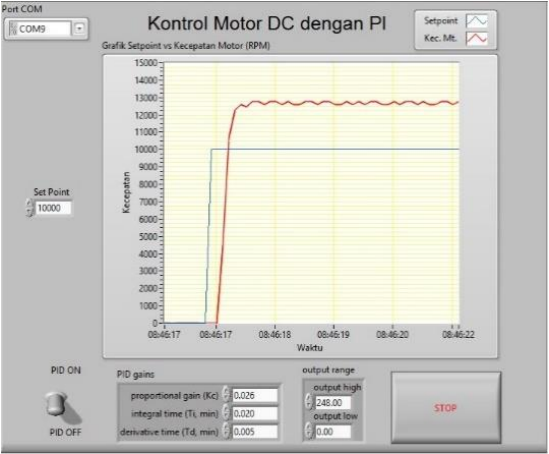


Gambar 4.6 *Set Point* pada LabVIEW



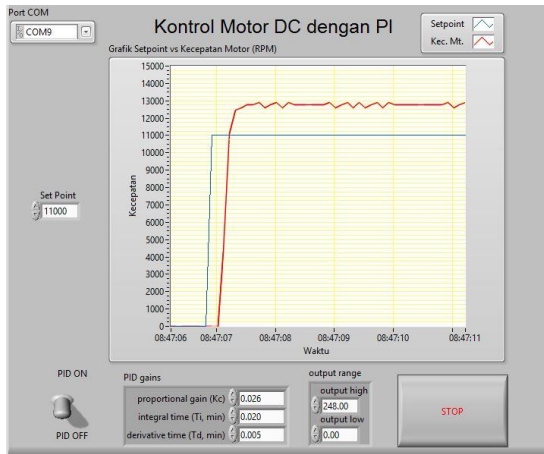
Gambar 4.7 *Range* PWM

Lalu menjalankan motor DC *Spindle* dengan memberikan nilai *Set Point* 10.000 rpm.



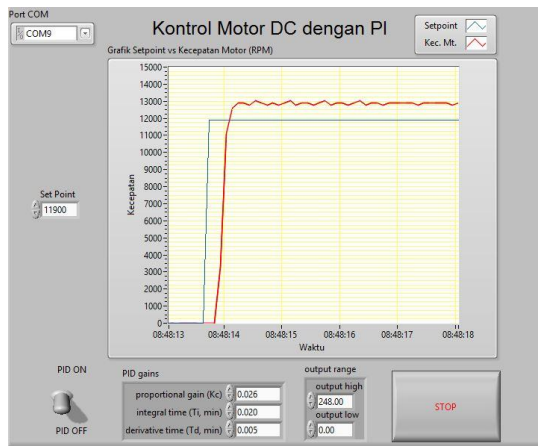
Gambar 4.8 *Set Point* 10.000 RPM

Kemudian menggerakkan motor DC *Spindle* dengan nilai *Set Point* 11.000 rpm.



Gambar 4.9 *Set Point* 11.000 RPM

Yang terakhir memberikan nilai *Set Point* 12.000 rpm dimana kecepatan tersebut adalah kecepatan maksimal.



Gambar 4.10 *Set Point* 12.000 RPM

4.2.2 Pengujian Pengaturan Kecepatan Motor dengan Kontroler PI

Pada pengujian kali ini dari nilai *Set Point* yang telah diberikan akan diberikan pengontrolan dengan kontroler PI agar nilai dari kecepatan motor DC *Spindle* sesuai dengan *Set Point*. Adapun parameter yang diberikan didalam pengontrolan menggunakan kontroler PI yaitu parameter K_p dan τ_i . Nilai dari parameter tersebut diberikan dengan nilai yang berbeda-beda sehingga dapat membandingkan antara parameter satu dengan yang lain dan bisa menentukan parameter yang sesuai untuk diterapkan pada pengontrolan kecepatan motor DC *Spindle* menggunakan kontroler PI. Parameter yang diberikan dapat dilihat pada Tabel di bawah ini.

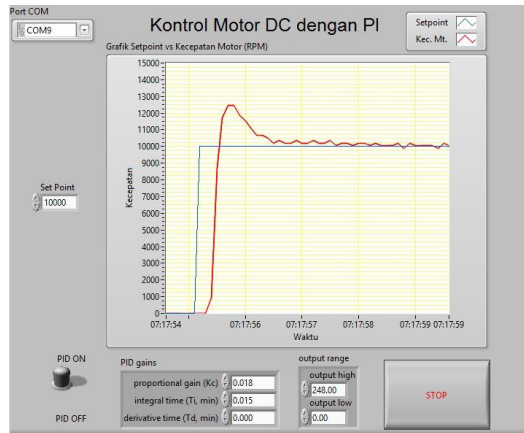
Tabel 4.5 Parameter Kontroler PI

No.	Kecepatan	Parameter		Kecepatan
		K_p	τ_i	
1.	10.000 RPM	0,018	0,015	Kurang Stabil
		0,012	0,011	Stabil
		0,026	0,020	Stabil
		0,037	0,032	Kurang Stabil
		0,047	0,046	Kurang Stabil
		0,058	0,055	Kurang Stabil
2.	11.000 RPM	0,018	0,015	Stabil
		0,012	0,011	Stabil
		0,026	0,020	Kurang Stabil
		0,037	0,032	Kurang Stabil
		0,047	0,046	Stabil
		0,058	0,055	Stabil
3.	12.000 RPM	0,018	0,015	Kurang Stabil
		0,012	0,011	Stabil
		0,026	0,020	Kurang Stabil
		0,037	0,032	Kurang Stabil
		0,047	0,046	Kurang Stabil
		0,058	0,055	Kurang Stabil

Pada Tabel diatas dapat diketahui bahwa nilai dari setiap parameter diberikan pada kecepatan 10.000 sampai 12.000 rpm. Dari situ dapat

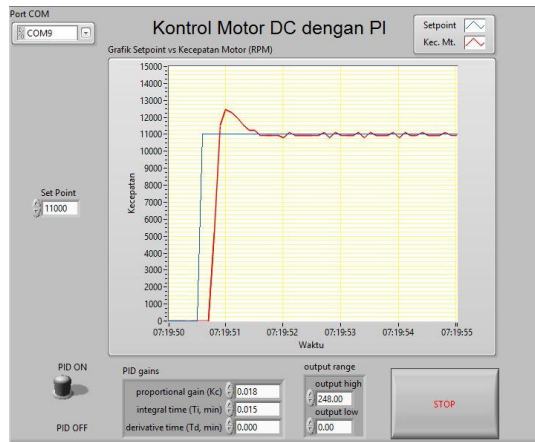
diketahui efek pengontrolan kecepatan oleh kontroler PI pada sistem motor DC *Spindle* yang menghasilkan respon yang Stabil.

Pada data pertama diberikan nilai *Set Point* 10.000 rpm dan diberikan nilai parameter $K_p = 0,018$ dan $\tau_i = 0.015$.



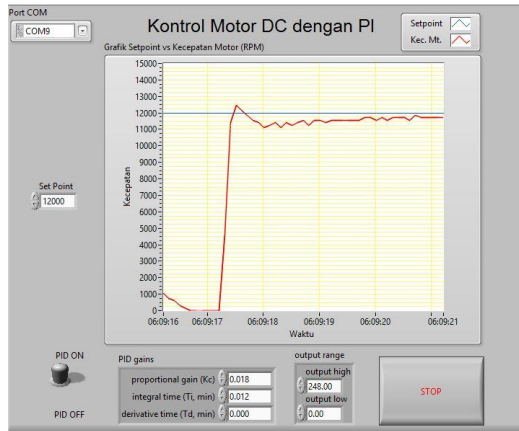
Gambar 4.11 Parameter $K_p = 0,018$ dan $\tau_i = 0.015$ pada *Set Point* 10.000 RPM

Pada data kedua diberikan nilai *Set Point* yang berbeda yaitu 11.000 RPM dengan parameter $K_p = 0,018$ dan $\tau_i = 0.015$.



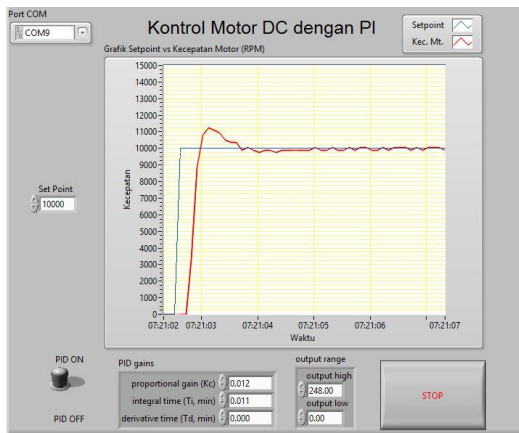
Gambar 4.12 Parameter $K_p = 0,018$ dan $\tau_i = 0.015$ pada *Set Point* 11.000 RPM

Pada data ketiga diberikan nilai *Set Point* yang berbeda yaitu 12.000 RPM dengan parameter $K_p = 0,018$ dan $\tau_i = 0.015$.



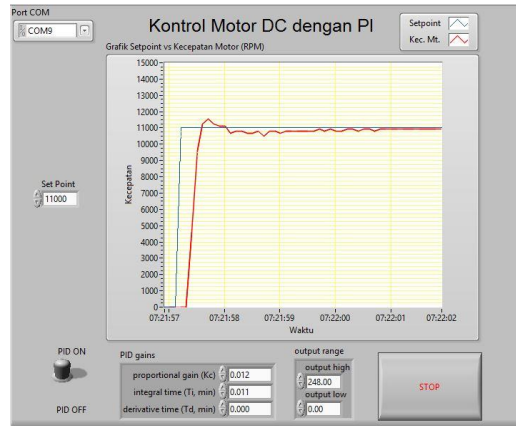
Gambar 4.13 Parameter $K_p = 0,018$ dan $\tau_i = 0.015$ pada *Set Point* 12.000 RPM

Pada data keempat diberikan nilai *Set Point* 10.000 RPM dengan parameter $K_p = 0,012$ dan $\tau_i = 0.011$.



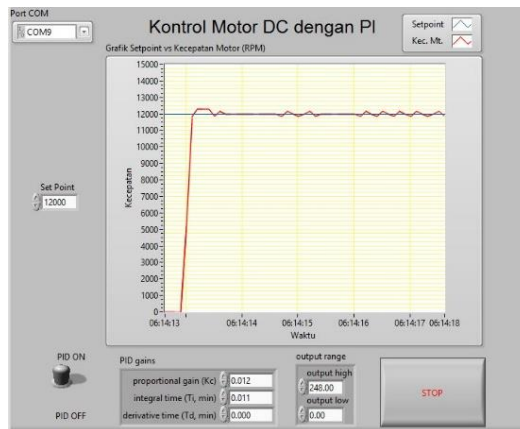
Gambar 4.14 Parameter $K_p = 0,012$ dan $\tau_i = 0.011$ pada *Set Point* 10.000 RPM

Pada data kelima diberikan nilai *Set Point* 11.000 RPM dengan parameter $K_p = 0,012$ dan $\tau_i = 0.011$.



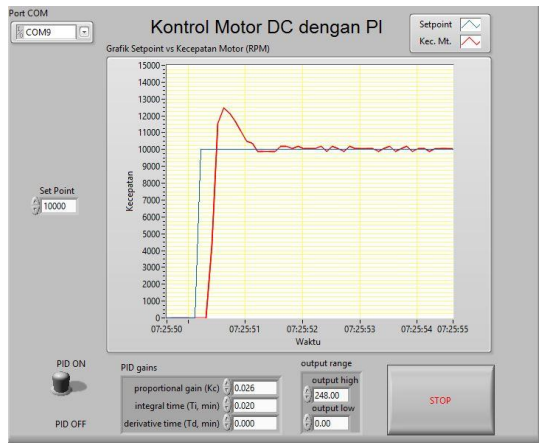
Gambar 4.15 Parameter $K_p = 0,012$ dan $\tau_i = 0.011$ pada *Set Point* 11.000 RPM

Pada data keenam diberikan nilai *Set Point* 12.000 RPM dengan parameter $K_p = 0,012$ dan $\tau_i = 0.011$.



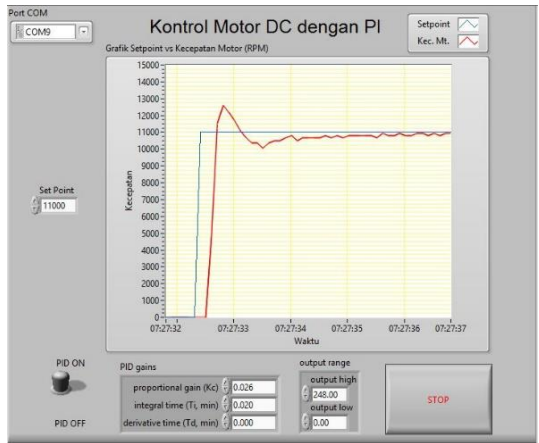
Gambar 4.16 Parameter $K_p = 0,012$ dan $\tau_i = 0.011$ pada *Set Point* 12.000 RPM

Pada data ketujuh diberikan nilai *Set Point* 10.000 RPM dengan parameter $K_p = 0,026$ dan $\tau_i = 0.020$.



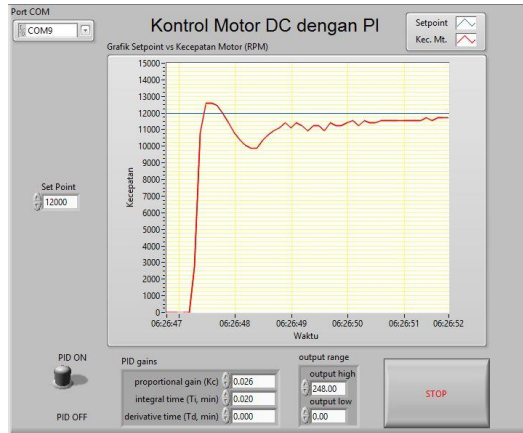
Gambar 4.17 Parameter $K_p = 0,026$ dan $\tau_i = 0.020$ pada *Set Point* 10.000 RPM

Pada data kedelapan diberikan nilai *Set Point* 11.000 RPM dengan parameter $K_p = 0,026$ dan $\tau_i = 0.020$.



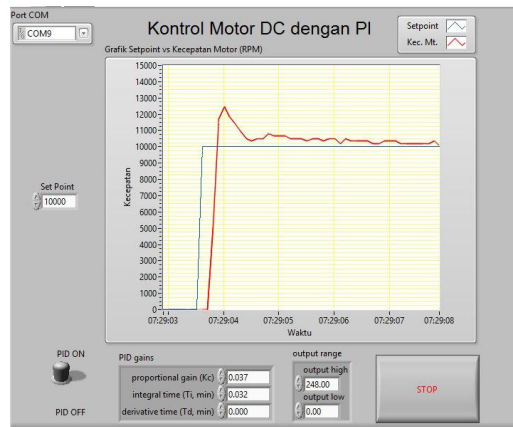
Gambar 4.18 Parameter $K_p = 0,026$ dan $\tau_i = 0.020$ pada *Set Point* 11.000 RPM

Pada data kesembilan diberikan nilai *Set Point* 12.000 RPM dengan parameter $K_p = 0,026$ dan $\tau_i = 0.020$.



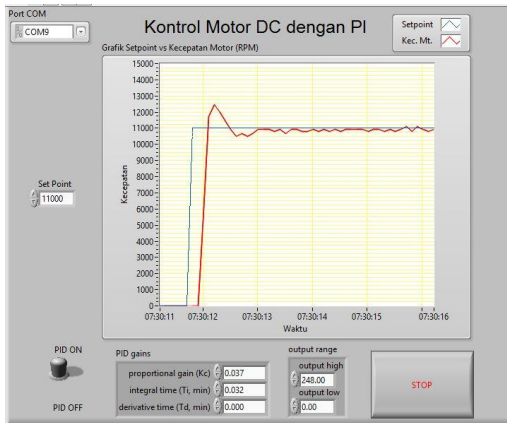
Gambar 4.19 Parameter $K_p = 0,026$ dan $\tau_i = 0.020$ pada *Set Point* 12.000 RPM

Pada data kesepuluh diberikan nilai *Set Point* 10.000 RPM dengan parameter $K_p = 0,037$ dan $\tau_i = 0.032$.



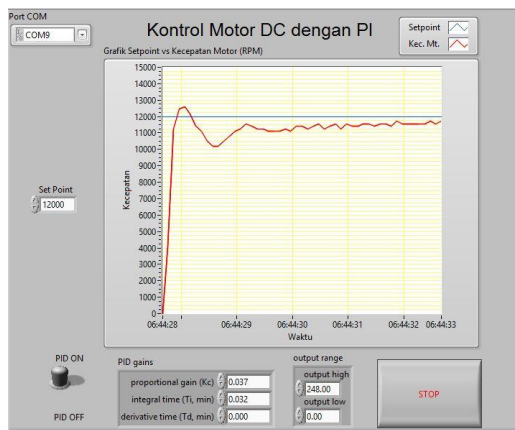
Gambar 4.20 Parameter $K_p = 0,037$ dan $\tau_i = 0.032$ pada *Set Point* 10.000 RPM

Pada data kesebelas diberikan nilai *Set Point* 11.000 RPM dengan parameter $K_p = 0,037$ dan $\tau_i = 0.032$.



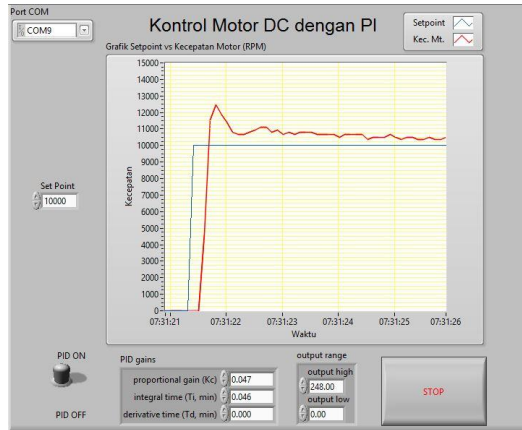
Gambar 4.21 Parameter $K_p = 0,037$ dan $\tau_i = 0.032$ pada *Set Point* 11.000 RPM

Pada data kedua belas diberikan nilai *Set Point* 12.000 RPM dengan parameter $K_p = 0,037$ dan $\tau_i = 0.032$.



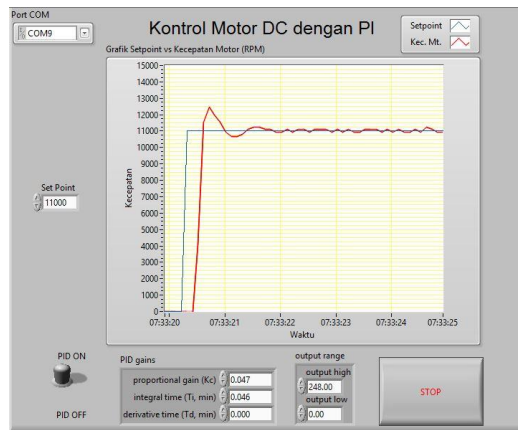
Gambar 4.22 Parameter $K_p = 0,037$ dan $\tau_i = 0.032$ pada *Set Point* 12.000 RPM

Pada data ketigabelas diberikan nilai *Set Point* 10.000 RPM dengan parameter $K_p = 0,047$ dan $\tau_i = 0.046$.



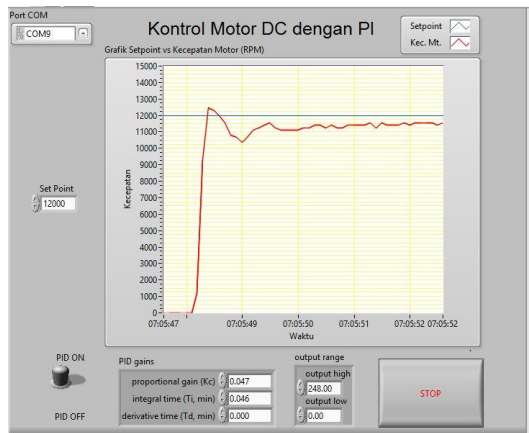
Gambar 4.23 Parameter $K_p = 0,047$ dan $\tau_i = 0.046$ pada *Set Point* 10.000 RPM

Pada data keempatbelas diberikan nilai *Set Point* 11.000 RPM dengan parameter $K_p = 0,047$ dan $\tau_i = 0.046$.



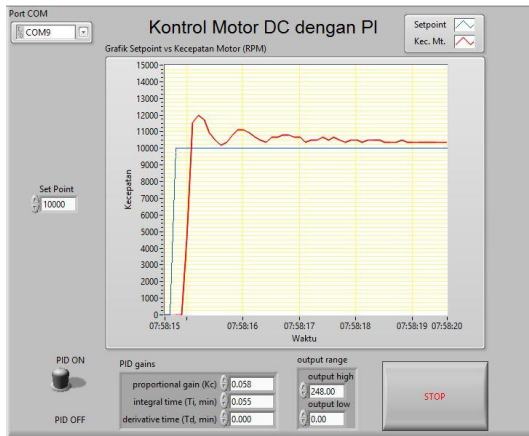
Gambar 4.24 Parameter $K_p = 0,047$ dan $\tau_i = 0.046$ pada *Set Point* 11.000 RPM

Pada data kelimabelas diberikan nilai *Set Point* 12.000 RPM dengan parameter $K_p = 0,047$ dan $\tau_i = 0.046$.



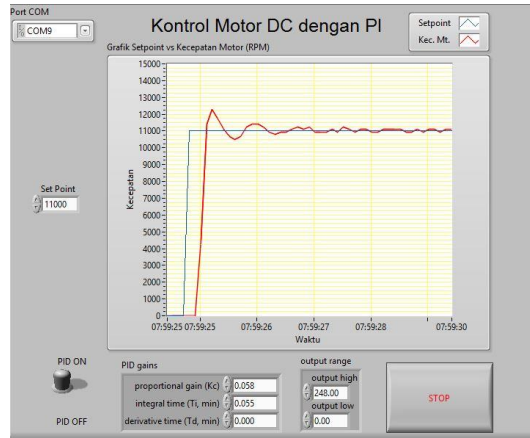
Gambar 4.25 Parameter $K_p = 0,047$ dan $\tau_i = 0.046$ pada *Set Point* 12.000 RPM

Pada data keenambelas diberikan nilai *Set Point* 10.000 RPM dengan parameter $K_p = 0,058$ dan $\tau_i = 0.055$.



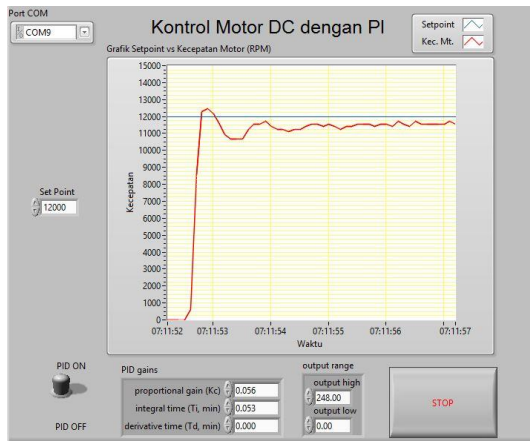
Gambar 4.26 Parameter $K_p = 0,058$ dan $\tau_i = 0.055$ pada *Set Point* 10.000 RPM

Pada data ketujuhbelas diberikan nilai *Set Point* 11.000 RPM dengan parameter $K_p = 0,058$ dan $\tau_i = 0.055$.



Gambar 4.27 Parameter $K_p = 0,058$ dan $\tau_i = 0.055$ pada *Set Point* 11.000 RPM

Pada data ketujuhbelas diberikan nilai *Set Point* 12.000 RPM dengan parameter $K_p = 0,058$ dan $\tau_i = 0.055$.



Gambar 4.28 Parameter $K_p = 0,058$ dan $\tau_i = 0.055$ pada *Set Point* 12.000 RPM

Dari seluruh hasil pengujian kontroler PI dapat disimpulkan bahwa parameter kontroler PI yang mengalami *overshoot* dan osilasi sedikit, juga waktu stabil kecepatan motor yang cepat dibandingkan parameter

yang lainnya adalah dengan nilai parameter $K_p = 0,012$ dan $\tau_i = 0.011$ karena menghasilkan respon motor yang bisa stabil dengan selang waktu 0,6 detik dari *Set Point* pada kecepatan maksimal 12.000 RPM.

4.2.3 Pengujian Pengaturan Kecepatan Motor dengan Kontroler PID

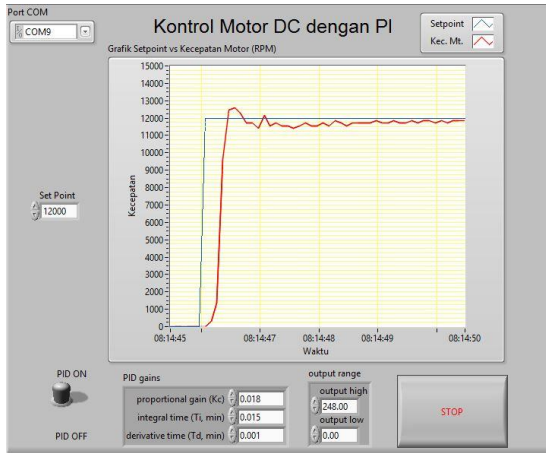
Pada pengujian kali ini dari nilai *Set Point* yang telah diberikan akan diberikan pengontrolan dengan kontroler PID agar nilai dari kecepatan motor DC *Spindle* sesuai dengan *Set Point*. Pengujian ini dilakukan dengan tujuan untuk membandingkan dengan kontroler PI manakah diantara kedua kontroler tersebut yang menghasilkan respon motor DC *Spindle* yang stabil. Berbeda dengan kontroler PI, pada kontroler PID ini akan diberikan parameter tambahan lagi selain parameter K_p dan τ_i yaitu parameter derivatif (τ_d). Parameter derivatif ini akan memberikan efek pengontrolan terhadap respon motor yaitu memperbaiki respon transien dengan memprediksi error yang terjadi.

Pengambilan data dilakukan dengan memberikan nilai parameter τ_d yang berbeda pada setiap parameter kontroler PI sehingga menghasilkan suatu kontroler PID dengan kecepatan maksimal yaitu 12.000 RPM. Parameter τ_d yang akan diberikan pada kontroler PI dapat dilihat pada Tabel di bawah ini.

Tabel 4.6 Parameter Kontroler PID

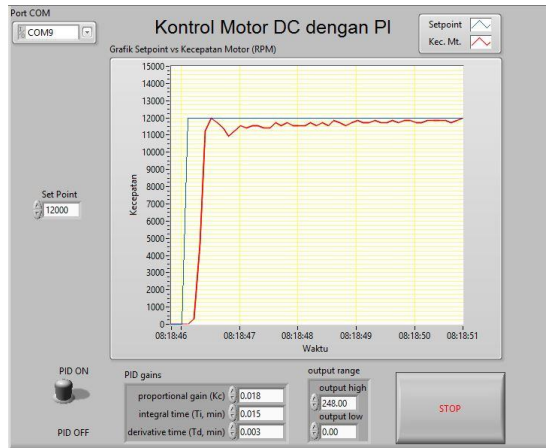
No.	Kecepatan	Parameter			Keterangan
		K_p	τ_i	τ_d	
1.	12.000 RPM	0,018	0,015	0,001	Kurang Stabil
2.	12.000 RPM	0,018	0,015	0,003	Stabil
3.	12.000 RPM	0,018	0,015	0,005	Kurang Stabil
4.	12.000 RPM	0,012	0,011	0,001	Stabil
5.	12.000 RPM	0,012	0,011	0,003	Kurang Stabil
6.	12.000 RPM	0,012	0,011	0,005	Kurang Stabil
7.	12.000 RPM	0,026	0,020	0,001	Kurang Stabil
8.	12.000 RPM	0,026	0,020	0,003	Kurang Stabil
9.	12.000 RPM	0,026	0,020	0,005	Kurang Stabil

Pada data pengujian pertama diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,001$.



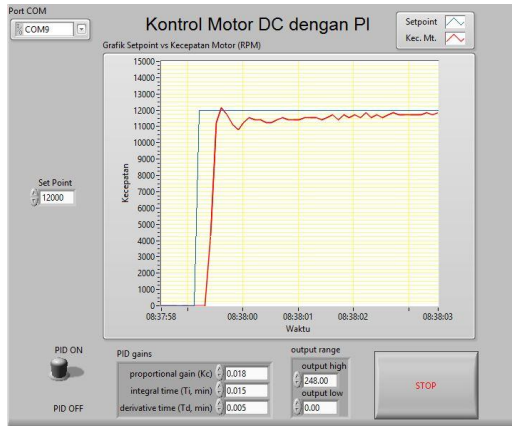
Gambar 4.29 Parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,001$ pada *Set Point* 12.000 RPM

Pada data pengujian kedua diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,003$.



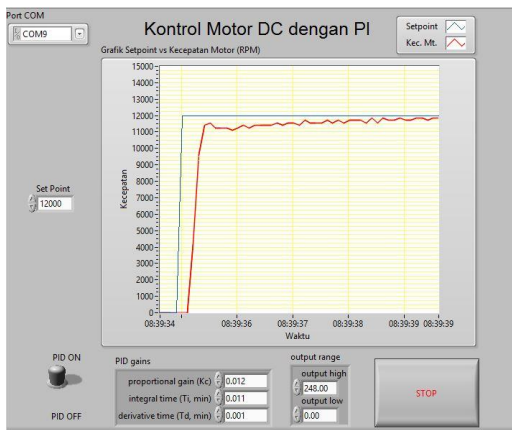
Gambar 4.30 Parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,003$ pada *Set Point* 12.000 RPM

Pada data pengujian ketiga diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,005$.



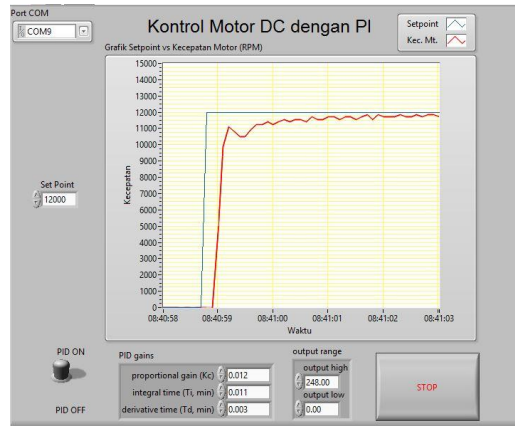
Gambar 4.31 Parameter $K_p = 0,018$, $\tau_i = 0,015$, dan $\tau_d = 0,005$ pada *Set Point* 12.000 RPM

Pada data pengujian keempat diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,001$.



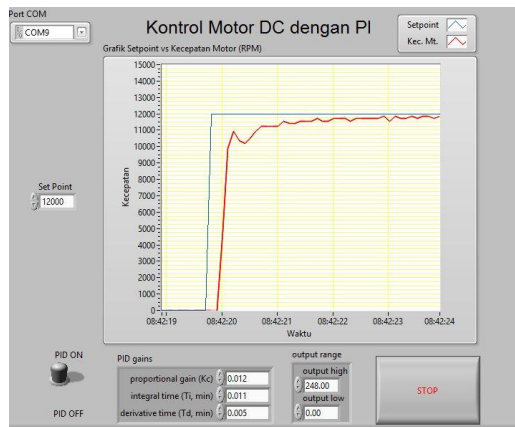
Gambar 4.32 Parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,001$ pada *Set Point* 12.000 RPM

Pada data pengujian kelima diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,003$.



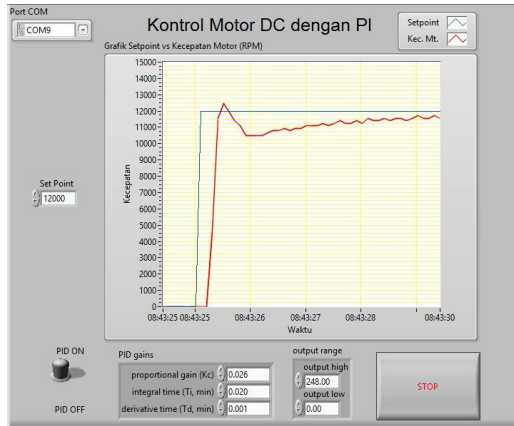
Gambar 4.33 Parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,003$ pada *Set Point* 12.000 RPM

Pada data pengujian keenam diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,005$.



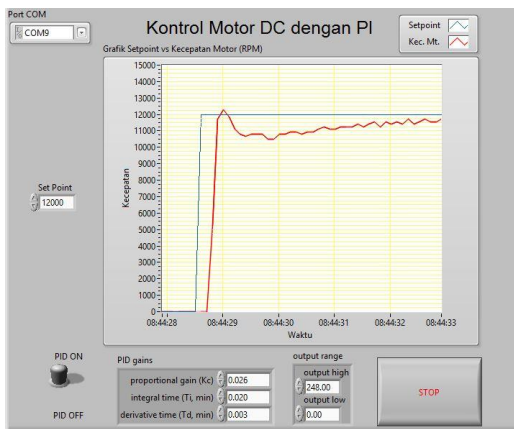
Gambar 4.34 Parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,005$ pada *Set Point* 12.000 RPM

Pada data pengujian ketujuh diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,001$.



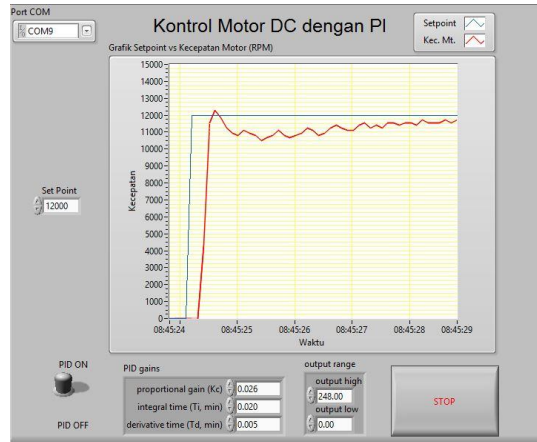
Gambar 4.35 Parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,001$ pada Set Point 12.000 RPM

Pada data pengujian kedelapan diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,003$.



Gambar 4.36 Parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,003$ pada Set Point 12.000 RPM

Pada data pengujian kesembilan diuji dengan kecepatan 12.000 RPM dengan parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,005$.

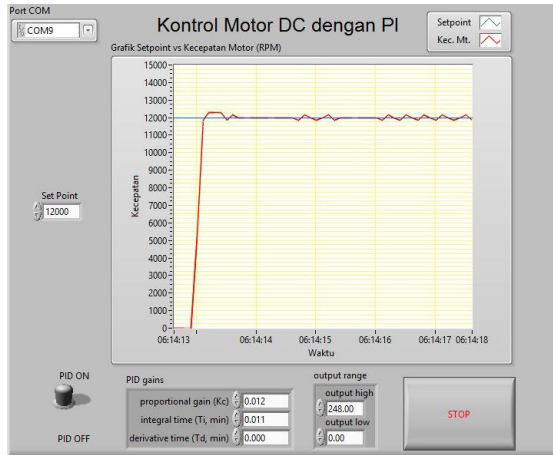


Gambar 4.37 Parameter $K_p = 0,026$, $\tau_i = 0,020$, dan $\tau_d = 0,005$ pada Set Point 12.000 RPM

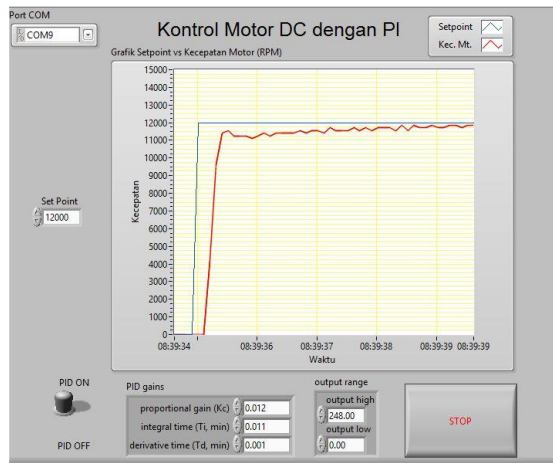
Dari seluruh hasil pengujian kontroler PID dapat disimpulkan bahwa parameter kontroler PID yang tidak terdapat *overshoot* dan osilasi adalah dengan nilai parameter $K_p = 0,012$, $\tau_i = 0,011$, dan $\tau_d = 0,001$ karena menghasilkan respon motor yang tidak melebihi dari Set Point yang diberikan dan bisa stabil kecepatan motor pada selang waktu 3,3 detik dari Set Point pada kecepatan maksimal 12.000 RPM.

4.2.4 Perbandingan Kontroler PI dan PID

Penggunaan kedua kontroler PI dan PID tentu memiliki efek pengontrolan yang berbeda terhadap respon motor DC *Spindle*. Karakteristik pengontrolan menggunakan kontroler PI akan memberikan penguatan pada respon motor DC *Spindle* sehingga terjadi *overshoot* dan kemudian menyetabilkan kecepatan motor DC *Spindle*. Pada kontroler PID akan memberikan efek pengontrolan yaitu menurunkan *overshoot* yang terjadi dan menyetabilkan kecepatan motor DC *Spindle* pada Set Point yang diberikan. Perbandingan antara kedua kontroler tersebut dapat dilihat pada Gambar 4.38 dan Gambar 4.39.



Gambar 4.38 Kontroler PI pada *Set Point* 12.000 RPM



Gambar 4.39 Kontroler PID pada *Set Point* 12.000 RPM

Dapat dilihat pada Gambar di atas bahwa dari kedua kontroler tersebut menghasilkan respon motor DC *Spindle* yang berbeda. Jika pada penggunaan kontroler PI respon yang dihasilkan terdapat *overshoot* dan bisa langsung stabil pada *Set Point* yang dihasilkan maka pada kontroler PID akan menurunkan *overshoot* yang terjadi namun membutuhkan waktu lebih untuk menyetabiilkan kecepatan motor DC *Spindle* pada *Set*

Point yang diberikan. Maka dari itu penggunaan kontroler PI lebih baik daripada menggunakan kontroler PID karena respon motor DC *Spindle* yang dihasilkan bisa langsung stabil dengan selang waktu 0,6 detik dari *Set Point* yang diberikan. Meskipun terdapat *overshoot* namun *overshoot* yang terjadi tidak terlalu berlebih dan setelah itu tidak terdapat osilasi pada respon motor DC *Spindle*.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari hasil pengujian yang telah dilakukan selama proses perancangan, pencarian data, hingga menganalisa data tersebut untuk Tugas Akhir ini, maka dapat diambil kesimpulan dan saran untuk perbaikan dan pengembangan alat TA kami kedepannya.

5.1 Kesimpulan

1. Metode pengontrolan menggunakan kontroler PI untuk mengatur kecepatan motor DC *Spindle* dapat digunakan dan terbukti tidak melebihi respon sistem yang diinginkan.
2. Pada kecepatan maksimal 12.000 RPM, kontroler PI mampu menstabilkan kecepatan motor DC *Spindle* sesuai dengan *Set Point* dalam selang waktu 0,6 detik dengan parameter kontrol $K_p = 0,012$ dan $\tau_i = 0.011$.
3. Penggunaan 2 motor *Stepper* pada sumbu Y dan penggunaan *fanbelt* membantu pergerakan dari mesin CNC *Portable* agar lebih presisi.
4. Penggunaan metode *Full Step* pada motor *Stepper* terbukti lebih cepat dalam pengerjaannya karena hanya memerlukan 4 kali langkah untuk mencapai derajat 180 dibandingkan dengan metode *Half Step* yang memerlukan 16 langkah untuk mencapai derajat 180.

5.2 Saran

Saran untuk penelitian dan pengembangan alat TA kami ini yaitu bisa dapat mengganti sensor kecepatan yang lebih akurat lagi agar data yang diperoleh semakin Stabil dan bisa didapatkan nilai *Plant* dari motor yang lebih akurat.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Arduino, <https://www.arduino.cc/>. Diakses pada tanggal 17 April 2017
- [2] Bitter, Rick. Taqi M. And Matt N. 2006. *LabVIEW : Advanced Programming Techniques*. New York : CRC Press.
- [3] Eko Ihsanto, Triwisaksana, Nacep Suryana. 2008. *Sistem Pengendali Motor DC Dengan PWM* (Jurnal). Jakarta: Universitas Mercu Buana
- [4] Elektronika Dasar, 2012. Prinsip Kerja Motor DC. <http://elektronika-dasar.web.id/prinsip-kerja-motor-dc/>. Diakses pada tanggal 17 April 2017
- [5] How To Mechatronics, 2015. How To Control a *Stepper* Motor with A4988 *Driver* and Arduino. <http://howtomechatronics.com/tutorials/arduino/how-to-control-Stepper-motor-with-a4988-Driver-and-arduino/>. Diakses pada tanggal 17 April 2017.
- [6] Kief, Hans B. And Waters, T. Frenderick. 1992. *Computer Numerical Control* (Journal). California: Glence
- [7] Permana, Fikri Yoga dan Rameli, Moch. 2013. *Pengaturan Kecepatan Spindle pada Retrofit Mesin Bubut CNC Menggunakan Kontroler PI Gain Scheduling* (Jurnal). Surabaya: Institut Teknologi Sepuluh Nopember
- [8] Recktenwald, Gerald. 2011. "Basic Pulse Width Modulation".
- [9] Reza, Selim., Al Mamun, Abdullah., Rahman, Atiar. And Begum, Mohaimina. 2015. *Design and Development of LabVIEW Based DC Motor Speed and Direction Control System* (Journal). Bangladesh: University of Dhaka
- [10] Saharkar. Nikita R. And Dhote. Girish M. 2013. *Prototype Development of Milling Machine Using CAD/CAM* (Journal). India: Dr. Babasaheb Ambedkar College of Engineering & Research
- [11] Yudistiro, Ardi 2011, PENERAPAN SENSOR OPTOCOUPLER PADA ALAT PENGUKUR KECEPATAN, Tugas Akhir Teknik Fisika, Universitas Negeri Malang.
- [12] Zona Elektro, 2015. Teori Motor *Stepper* : Jenis Dan Prinsip Motor *Stepper*. <http://zonaelektro.net/motor-Stepper/>. Diakses pada tanggal 17 April 2017

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

Program Pembacaan Sensor

```
volatile int rpmcount = 0;
int rpm = 0;
unsigned long lastmillis = 0;
int encoder_pin = 2;

void setup(){
  pinMode(encoder_pin,INPUT);
  pinMode(3,OUTPUT);
  Serial.begin(9600);
  attachInterrupt(0, rpm_fan, FALLING);
}

void loop(){
  if(Serial.available()){
    int a=Serial.parseInt();
    if(Serial.read()==char(13)) analogWrite(3,a);
  }
  if (millis() - lastmillis == 100){
    detachInterrupt(0);
    Serial.println(rpmcount);
    rpmcount = 0;
    lastmillis = millis();
    attachInterrupt(0, rpm_fan, FALLING);
  }
}

void rpm_fan(){
  rpmcount++;
}
```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

B.1. Datasheet Driver Motor Stepper A4988



A4988

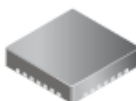
DMOS Microstepping Driver with Translator and Overcurrent Protection

Features and Benefits

- Low $R_{DS(on)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $1/2$, $1/4$, $1/8$, and $1/16$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(RT package)



Approximate size

Description

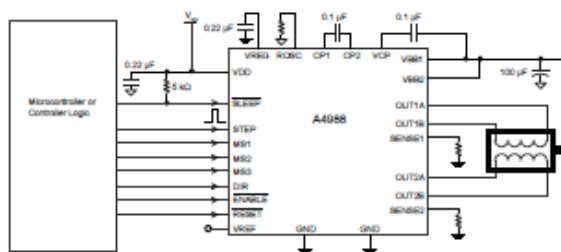
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ± 2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page...

Typical Application Diagram



A4988

DMOS Microstepping Driver with Translator and Overcurrent Protection

Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix -T), with 100% matte tin plated leadframes.

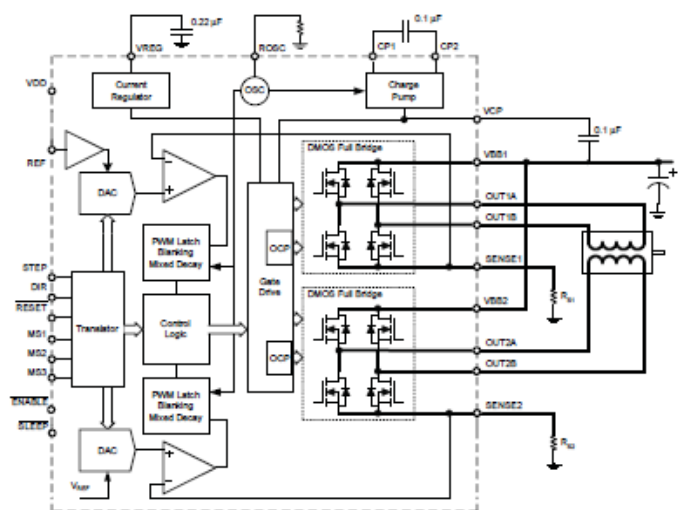
Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{DD}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		-0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		-0.3 to 5.5	V
VBx to OUTx			35	V
Sense Voltage	V_{SDS}		0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range 5	-20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	T_{STG}		-55 to 150	°C

Functional Block Diagram

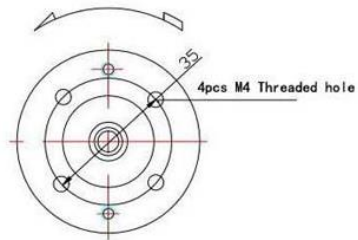
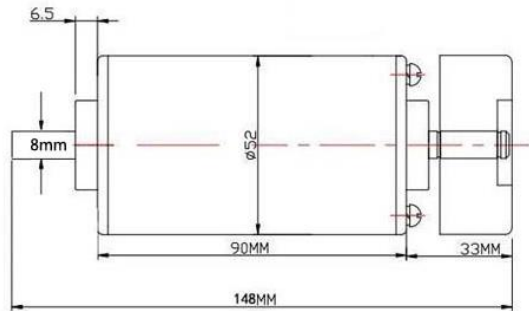


ELECTRICAL CHARACTERISTICS¹ at $T_A = 25^\circ\text{C}$, $V_{DD} = 3.5\text{ V}$ (unless otherwise noted)

Characteristic	Symbol	Test Conditions	Min.	Typ. ²	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{DS}	Operating	8	—	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	—	5.5	V
Output On Resistance	$R_{DS(on)}$	Source Driver, $I_{OUT} = -1.5\text{ A}$	—	320	430	mΩ
		Sink Driver, $I_{OUT} = 1.5\text{ A}$	—	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, $I_F = -1.5\text{ A}$	—	—	1.2	V
		Sink Diode, $I_F = 1.5\text{ A}$	—	—	1.2	V
Motor Supply Current	I_{DS}	$f_{SW} < 50\text{ kHz}$	—	—	4	mA
		Operating, outputs disabled	—	—	2	mA
Logic Supply Current	I_{DD}	$f_{SW} < 50\text{ kHz}$	—	—	8	mA
		Outputs off	—	—	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		$V_{DD} \times 0.7$	—	—	V
	$V_{IN(2)}$		—	—	$V_{DD} \times 0.3$	V
Logic Input Current	$I_{IN(1)}$	$V_{IN} = V_{DD} \times 0.7$	-20	<1.0	20	μA
	$I_{IN(2)}$	$V_{IN} = V_{DD} \times 0.3$	-20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	—	100	—	kΩ
	R_{MS2}	MS2 pin	—	50	—	kΩ
	R_{MS3}	MS3 pin	—	100	—	kΩ
Logic Input Hysteresis	$V_{IHYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	μs
		$R_{OSC} = 25\text{ k}\Omega$	23	30	37	μs
Reference Input Voltage Range	V_{REF}		0	—	4	V
Reference Input Current	I_{REF}		-3	0	3	μA
Current Trip-Level Error ³	ϵ_{IL}	$V_{REF} = 2\text{ V}$, $\%I_{TRIP(max)} = 38.27\%$	—	—	±15	%
		$V_{REF} = 2\text{ V}$, $\%I_{TRIP(max)} = 70.71\%$	—	—	±5	%
		$V_{REF} = 2\text{ V}$, $\%I_{TRIP(max)} = 100.00\%$	—	—	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold	$I_{OCP(ST)}$		2.1	—	—	A
Thermal Shutdown Temperature	T_{SD}		—	165	—	°C
Thermal Shutdown Hysteresis	T_{SDHYS}		—	15	—	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		—	90	—	mV

¹For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.²Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.³ $\%V_{ERR} = [(V_{ASD(8)} - V_{ASD(8)}) / (V_{ASD(8)})]$.

B.2. Datasheet Motor DC Spindle



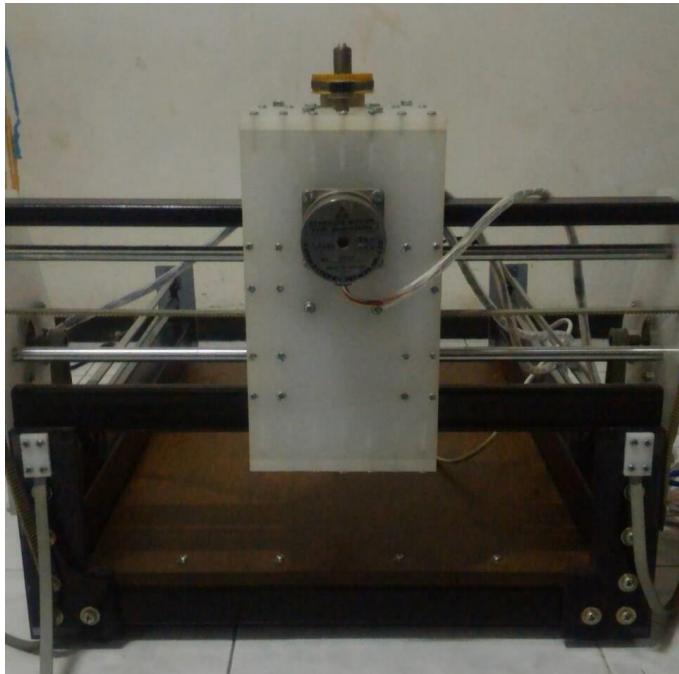
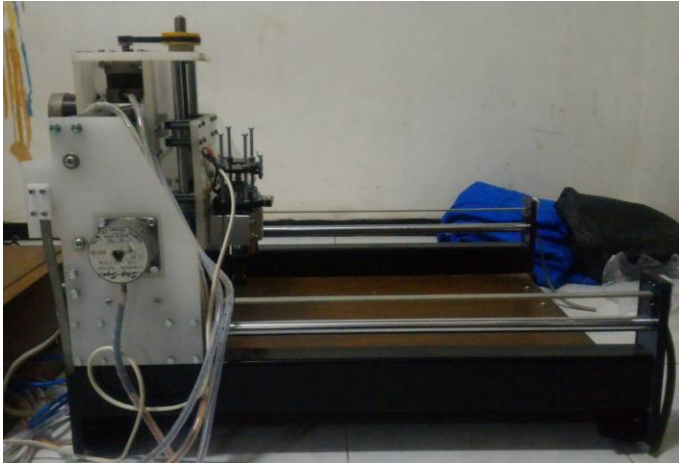
<i>Spindle Motor</i>	Brushless DC Motor
Tegangan	12-48 V DC
Daya	300 Watt
Kecepatan	3000-12000r/min (12 V-3000 putaran 24 V-6000 putaran 36 V-9000 putaran 48 V-12000 putaran)
Torsi	400mN.m
Resistansi Insulalasi	>2 megaohms
Kekuatan <i>Dielectric</i>	400V
Diameter	52mm

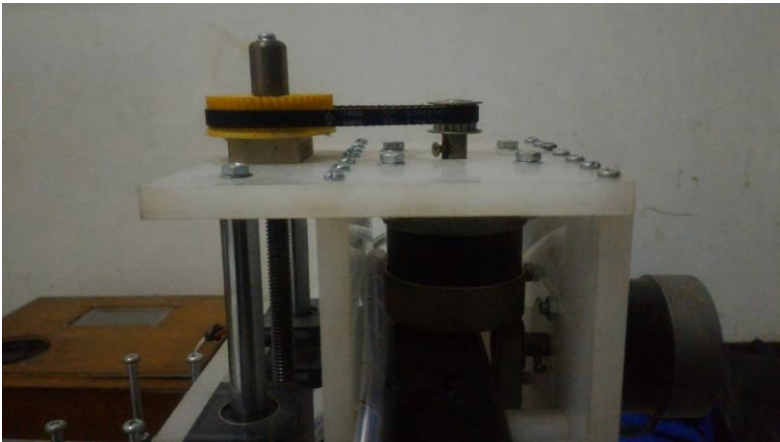
Panjang collet	35mm
Diameter collet	16mm
Panjang Motor	175mm (termasuk penjepit motor dan motor)
<i>Spindle radial runout</i>	Rentang 0.01-0.04

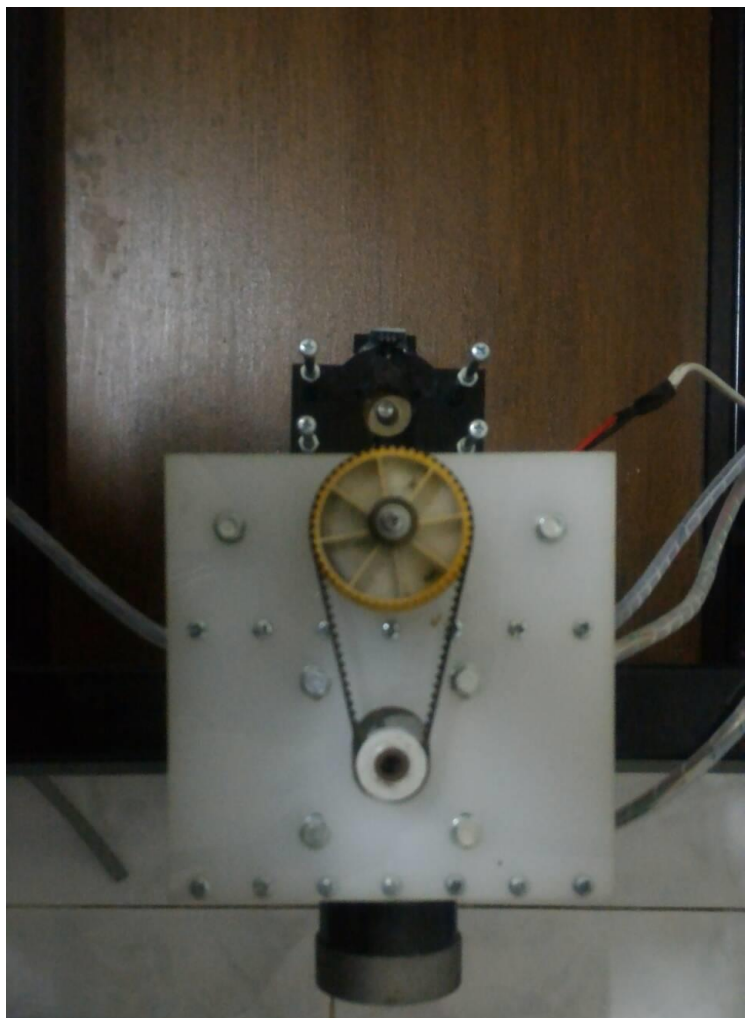
LAMPIRAN C

Tampilan Mesin CNC *Portable*











-----Halaman ini sengaja dikosongkan-----



DAFTAR RIWAYAT HIDUP

Nama : Rico Artha Dinata
TTL : Surabaya, 16 Juni 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Jalan Asemrowo I/7
Surabaya
Telp/HP : 081330351212
E-mail : ricoartha3@gmail.com

RIWAYAT PENDIDIKAN

- 2002 – 2008 : SDN Gunong Sekar 1 Sampang
- 2008 – 2011 : SMPN 2 Surabaya
- 2011 – 2014 : SMAN 9 Surabaya
- 2014 – 2017 : Bidang Studi Komputer Kontrol,
Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di Balai Riset dan Standardisasi Industri Surabaya
(Januari - Februari 2017).

PENGALAMAN KEPANITIAAN

- Organizing Comittee GERIGI ITS 2015.
- Kakak Pendamping GERIGI ITS 2016.
- Organizing Comittee LKMM Pra TD XIII FTI ITS 2015.